

Slackware for dummies

Mauro Sacchetto

3° edizione: ottobre 2006

Copyright © 2006 Mauro Sacchetto

È garantito il permesso di copiare, distribuire e/o modificare questo documento seguendo i termini della Licenza per Documentazione Libera GNU, Versione 1.1 o ogni versione successiva pubblicata dalla Free Software Foundation.

Una copia della licenza è acclusa nella sezione intitolata «The GNU General Public License».

Indice

Introduzione	1
Premessa	1
Un cenno storico	2
Requisiti	3
Informazioni su Slackware	3
1 Procurarsi Slackware	5
1.1 Scaricare Slackware	5
Scheda n. 1: Condivisioni con BitTorrent	5
1.2 Controllare le ISO	7
1.2.1 Quali controlli	7
1.2.2 Lo md5sum	7
1.2.3 La firma digitale	7
1.2.4 Controllo dei file	8
Scheda n. 2: GnuPG, firma e crittografia	8
1.3 Masterizzare le ISO	10
1.3.1 Masterizzare le ISO su CD col kernel 2.4	10
1.3.2 Masterizzare le ISO su CD col kernel 2.6	10
1.3.3 Controllo del CD	10
1.3.4 Creare ISO da CD	10
1.3.5 Reperire Slackware in DVD	11
1.3.6 Masterizzare le ISO su DVD	11
1.3.7 I front-end grafici	11
2 Preparare il disco fisso	13
2.1 Partizionare il disco fisso	13
2.1.1 Partizioni richieste	13
2.1.2 Due casi esemplificativi	13
2.2 Usare Cfdisk	14
2.2.1 Fdisk e Cfdisk	14
2.2.2 Creare le partizioni	14
2.2.3 Formattare le partizioni	19
Scheda n. 3: I file system di GNU/Linux	19
2.2.4 Il file di swap	22
2.2.5 Le partizioni DOS	22
2.2.6 Una partizione di scambio	22
2.2.7 Esempi di configurazione	23
Scheda n. 4: La partizione /home	23

3	Installare Slackware	25
3.1	Avviare l'installazione	25
	Scheda n. 5: I kernel di Slackware 11.0	25
3.2	Scegliere la tastiera	27
3.3	Lanciare il setup	28
3.4	Opzione ADDSWAP	29
3.5	Opzione TARGET	30
3.6	Opzione SOURCE	33
3.7	Opzione SELECT	34
	Scheda n. 6: I pacchetti di Slackware 11.0	34
3.8	Rifinire l'installazione	40
3.8.1	Scegliere un kernel	40
3.8.2	Creare un floppy di avvio	41
	Scheda n. 7: La gestione dei floppy	42
3.8.3	Configurare il modem	42
3.8.4	Abilitare Hotplug	43
3.8.5	Installare Lilo	43
3.8.6	Configurare il mouse	45
3.8.7	Abilitare GPM	46
3.8.8	Configurare la rete	46
3.8.9	Impostare i demoni	47
3.8.10	Sostituire i font della console	48
3.8.11	Impostare l'ora di sistema	49
3.8.12	Scegliere un Desktop Environment	50
	Scheda n. 8: Come installare Gnome	51
3.8.13	Impostare la password di root	54
3.8.14	Riavviare il sistema	55
3.9	Installare Slackware da floppy	56
	Scheda n. 9: Le directory di Slackware 11.0	56
4	Avviare Slackware per la prima volta e spegnerla	59
4.1	Avviare Slackware	59
4.1.1	Entrare nel sistema	59
4.1.2	Leggere i messaggi di sistema	59
4.1.3	Avviare un Window manager	59
4.2	Italianizzare Slackware	60
4.2.1	Impostare la tastiera	60
4.2.2	Italianizzare KDE	60
4.2.3	Italianizzare altre applicazioni	60
4.3	Spegnere Slackware	61
4.3.1	Uscire da una sessione	61
4.3.2	Spegnere il computer da root	61
4.3.3	Spegnere fisicamente il computer	61
4.3.4	Spegnere il computer da utente	62

5	Ottimizzare il server grafico	65
5.1	Funzionamento del server grafico	65
5.2	Gli strumenti di configurazione	66
5.2.1	Usare xorgconfig	66
5.2.2	Come modificare xorg.conf	69
	Scheda n. 10: L'editor Vi	69
5.3	Ricorrere a Xvidtune	74
5.4	Abilitare la rotellina del mouse	75
5.5	Abilitare l'accelerazione grafica nativa	75
5.6	Installare i driver per l'accelerazione grafica (schede ATI)	76
5.6.1	Scaricare i driver	76
5.6.2	Installazione manuale del driver	76
5.6.3	Installazione automatica del driver	77
5.6.4	Verifica dell'installazione	78
5.6.5	Errori eventuali	78
5.7	Installare i driver per l'accelerazione grafica (schede Nvidia)	79
5.7.1	Scaricare e installare il driver	79
5.7.2	Configurare xorg.conf	79
	Scheda n. 11: Il server X e il file /etc/X11/xorg.conf	80
5.8	Esportare il server X	83
5.8.1	Server e client	83
5.8.2	Impostare il server	83
5.8.3	Impostare il client	83
5.8.4	Lanciare i programmi	84
5.8.5	Applicazioni grafiche da root in sessione utente	84
	Scheda n. 12: L'editor GNU Emacs	84
6	Ottimizzare il server sonoro	89
6.1	Configurare il server sonoro	89
	Scheda n. 13: La gestione dei moduli	89
	Scheda n. 14: Il sistema sonoro in GNU/Linux	91
6.2	Il server sonoro e gli utenti	93
6.3	Ascoltare i file MIDI	93
6.3.1	Che cosa sono i file MIDI	93
6.3.2	Ascoltare i MIDI con schede dotate di sintetizzatore MIDI	94
6.3.3	Ascoltare i MIDI con schede prive di sintetizzatore MIDI	95
	Scheda n. 15: Che cosa sono i permessi	97
7	Gestione degli utenti e dei gruppi	101
7.1	Gestione degli utenti	101
7.1.1	Creare gli utenti	101
7.1.2	Rimuovere gli utenti	101
7.1.3	Disabilitare gli utenti	102
7.1.4	Gestire le quote degli utenti	102
7.2	Le password	103
7.2.1	Il sistema delle password in Slackware	103
7.2.2	Se si dimentica la password di root	104
7.2.3	Se si dimentica la password di utente	104
7.3	Gestione dei gruppi	105
7.3.1	I gruppi	105

7.3.2	Creare un nuovo gruppo	105
7.3.3	Aggiungere utenti a un gruppo	105
8	Gestione delle unità	107
8.1	I dispositivi in GNU/Linux	107
8.1.1	Aggiungere dispositivi	107
8.1.2	I tipi di dispositivo	107
8.1.3	Creare nuovi dispositivi	108
8.2	Devfs e udev	108
8.2.1	Il kernel 2.4 e devfs	108
8.2.2	Il kernel 2.6 e udev	109
8.2.3	La configurazione di udev	110
8.2.4	La sintassi di udev	111
8.2.5	L'identificazione dei dispositivi	111
8.2.6	Creare regole per udev	114
8.2.7	Alcuni esempi di regole per udev	114
	Scheda n. 16: I link simbolici	116
8.3	Il montaggio dei dispositivi	117
8.3.1	Il file /etc/fstab	117
8.3.2	Montaggio dei dispositivi presenti in /etc/fstab	117
8.3.3	Montaggio dei dispositivi non presenti in /etc/fstab	117
	Scheda n. 17: Il file /etc/fstab	117
8.4	Montaggio dei dischi rigidi	121
8.5	Montaggio di lettori CD e masterizzatori o unità DVD	121
8.6	Montaggio di un pendrive USB	122
	Scheda n. 18: Automount e autofs	122
9	L'installazione delle periferiche	127
9.1	Installare una stampante	127
9.1.1	Avviare Cups	127
9.1.2	Installare una stampante USB	127
9.1.3	Installare una stampante parallela	128
	Scheda 19: I server di stampa di GNU/Linux	128
9.1.4	Condividere una stampante tra sistemi GNU/Linux	129
9.1.5	Condividere una stampante tra GNU/Linux e Windows	131
9.2	Installare uno scanner USB	132
9.2.1	Kernel 2.4	132
9.2.2	Kernel 2.6	132
9.2.3	Condividere uno scanner	133
9.3	Installare una webcam	134
9.3.1	La webcam	134
9.3.2	Gnomemeeting	135
9.4	Installare una macchina fotografica digitale	135
9.4.1	Le macchine fotografiche digitali come dischi USB	136
9.4.2	Il protocollo PTP	136

10 Avviare il sistema e creare un dual boot	139
10.1 Il dual boot	139
10.2 Lilo	139
10.2.1 Il bootloader Lilo	139
10.2.2 Modalità di installazione di Lilo	139
10.2.3 La sintassi di Lilo	140
Scheda n. 20: Un esempio di /etc/lilo.conf	141
10.2.4 La password di Lilo	143
10.2.5 Aggiungere Windows	144
10.2.6 Riconfigurare Lilo	144
10.2.7 Il multiboot	144
10.2.8 Reinstallazione di Lilo	145
10.3 L'alternativa a Lilo: Grub	145
10.3.1 Il bootloader Grub	145
10.3.2 Modalità di installazione	146
10.3.3 La sintassi di Grub	146
10.3.4 Le voci di Grub	147
Scheda n. 21: Un esempio di /boot/grub/menu.lst	148
10.3.5 La password di Grub	149
10.3.6 Reinstallazione di Grub	149
10.4 Boot manager indipendenti	150
11 Gestione dei pacchetti	153
11.1 I pacchetti .tgz	153
Scheda n. 22: Le funzioni di Pkgtool	153
11.2 Il comando installpkg	154
11.3 Il comando upgradepkg	154
11.4 Il comando removepkg	155
11.5 Il comando explodepkg	155
11.6 Il comando makepkg	155
11.7 La gestione delle dipendenze	155
12 Creazione di pacchetti .tgz	157
12.1 La creazione manuale dei pacchetti	157
Scheda n. 23: Gli strumenti di compilazione	157
12.2 Procedura per la creazione di pacchetti .tgz	159
12.2.1 Scompattare il sorgente	160
Scheda n. 24: La gestione degli archivi	160
12.2.2 Sistemare i permessi	162
12.2.3 Esaminare le opzioni di configurazione	162
12.2.4 Configurare il pacchetto	167
12.2.5 Compilare il pacchetto	167
12.2.6 Installare il pacchetto	168
12.2.7 Aggiungere la documentazione	168
12.2.8 Strappare i binari	168
12.2.9 Strappare le librerie	169
12.2.10 Comprimere i file man	169
12.2.11 Comprimere i file info	169
12.2.12 Ripristinare i permessi	169
12.2.13 Inserire la descrizione del pacchetto e altri eventuali file informativi	170

12.2.14 Creare il pacchetto	171
12.3 Casi particolari	172
12.4 Lo script doinst.sh	172
12.5 Tips&tricks	173
12.5.1 Pacchetti con nomi non canonici	173
12.5.2 Programmi da avviarsi mediante uno script rc	174
12.5.3 Pacchetti che non compilano tutto	175
12.6 Creare una patch	175
Scheda n. 25: Lo streaming editor SED	176
12.7 Velocizzare le ricompilazioni	178
12.8 Il «fake install»	179
12.9 Compilazione, kernel e librerie	179
12.10 Lo SlackBuild	180
12.10.1 Che cos'è uno SlackBuild	180
12.10.2 Analisi di uno SlackBuild	180
12.10.3 Tips&tricks per SlackBuild	183
12.11 Creare pacchetti .tgz con Checkinstall	184
12.12 Creare pacchetti .tgz con Slacktrack	185
12.13 Usare gli .rpm	185
13 Aggiornare Slackware	187
13.1 Regole generali	187
13.2 Swaret	187
13.2.1 Che cos'è Swaret	187
13.2.2 Configurare Swaret	188
13.2.3 Aggiornare con Swaret	189
13.3 Slapt-get	191
13.4 Emerde	192
13.5 Slackupdate	192
13.6 Aggiornare la versione precedente alla successiva	192
13.7 Aggiornare alla current	193
13.8 Creare un DVD della current	194
14 Connettività e reti	195
14.1 Che cosa sono le reti	195
14.1.1 Le reti informatiche	195
14.1.2 I protocolli di comunicazione	195
14.1.3 Il protocollo TCP/IP	196
14.1.4 L'indirizzo IP	196
14.1.5 Indirizzi speciali	199
14.1.6 Indirizzi pubblici e privati	199
14.1.7 Comunicazione fra reti e sottoreti	200
14.1.8 La risoluzione dei nomi	200
14.1.9 Risoluzione statica e dinamica	200
14.2 I dispositivi di rete	201
14.2.1 Informazioni sui dispositivi	201
14.2.2 Configurazione di un dispositivo di rete	202
14.2.3 L'utilità netconfig	205
14.2.4 Configurare il routing	208
14.2.5 Configurare un server DHCP/DNS	209

14.2.6	Configurare una rete con KDE	213
14.3	File di configurazione della rete	213
14.3.1	Il file /etc/hosts	214
14.3.2	Il file /etc/resolv.conf	214
14.3.3	Il file /etc/host.conf	215
14.3.4	Il file /etc/rc.d/rc.inet.1.conf	216
14.3.5	I file /etc/hosts.allow e /etc/hosts.deny	216
14.3.6	Il file /etc/HOSTNAME	217
14.3.7	Il file /etc/networks	218
14.3.8	Il file /etc/protocols	218
14.3.9	Il file /etc/services	218
14.4	Comandi per la configurazione della rete	218
14.4.1	Il comando ping	218
14.4.2	Il comando traceroute	219
14.4.3	Il comando host	220
14.4.4	Il comando dig	221
14.4.5	Il comando nslookup	222
14.4.6	Il comando whois	222
14.4.7	Il comando finger	224
14.4.8	Il comando telnet	224
14.4.9	Il comando ssh	224
14.5	Connettersi a Internet	224
14.5.1	Gli Winmodem	224
14.5.2	Connessione con un modem esterno 56KB	225
14.5.3	Connettersi a Internet con un modem ADSL	227
14.5.4	Connettersi con un modem USB	227
14.5.5	Connettersi con un modem Ethernet	228
14.5.6	Connettersi a Internet con un router	228
14.6	Impostare un firewall con Guarddog	229
14.6.1	Linux e il firewall	229
14.6.2	La linguetta Zona	229
14.6.3	La linguetta Protocollo	230
14.6.4	La linguetta Accesso	230
14.6.5	La linguetta Avanzato	231
15	Condivisioni con NFS	233
15.1	Connettere i computer	233
15.2	Il protocollo NFS	233
15.3	Configurazione del server	234
15.4	Configurazione del client	236
15.5	La risoluzione dei nomi	237
15.6	Messaggi di errore	237
15.7	Il controllo degli accessi	237
15.8	Problemi di sicurezza	238
16	Condivisioni con Samba	239
16.1	Che cos'è Samba	239
16.2	Come funziona Samba	240
16.3	Come predisporre Windows per Samba	241
16.4	Il file di configurazione /etc/samba/smb.conf	242

16.4.1	Aspetti generali della configurazione	242
16.4.2	La sezione [global]	243
16.4.3	La sezione [homes]	250
16.4.4	La sezione [printers]	250
16.4.5	La sezione [tmp]	251
16.4.6	La sezione [public]	251
16.5	Alcuni esempi di /etc/samba/smb.conf	252
16.5.1	Configurazione di base	252
16.5.2	Condivisione delle home degli utenti	253
16.5.3	Condivisione pubblica in sola lettura	253
16.5.4	Condivisione pubblica in lettura e scrittura	254
16.5.5	Una directory accessibile a tutti gli utenti in lettura e scrittura	254
16.5.6	Una directory accessibile indiscriminatamente a tutti gli utenti in lettura e scrittura	255
16.5.7	Condivisione sicura in lettura e scrittura	255
16.5.8	Una directory accessibile solo ad alcuni utenti	256
16.5.9	Una directory privata con permessi preimpostati	256
16.5.10	Condivisione di un CD-Rom	257
16.6	Le utilità di Samba	257
16.6.1	L'utilità smbstatus	258
16.6.2	L'utilità smbtree	259
16.6.3	L'utilità findsmb	259
16.6.4	L'utilità smbpasswd	260
16.6.5	L'utilità testparm	260
16.6.6	L'utilità smbcontrol	260
16.6.7	L'utilità nmblookup	261
16.6.8	L'utilità pdbedit	261
16.6.9	L'utilità smbclient	262
16.6.10	L'utilità smbmount	263
16.6.11	L'utilità smbmount	263
16.6.12	Altre tecniche di montaggio e smontaggio	263
16.7	La stampa con Samba	264
16.7.1	La configurazione di /etc/samba/smb.conf per la stampa	264
16.7.2	Alcuni esempi di configurazione	265
16.7.3	Stampare da Windows con stampanti installate su sistemi GNU/Linux	267
16.7.4	Stampare da GNU/Linux con stampanti installate su sistemi Windows	268
16.8	La configurazione di Samba mediante SWAT	269
16.8.1	Abilitare SWAT	269
16.8.2	L'interfaccia di SWAT	270
17	La console	273
17.1	Tipi di console	273
17.2	Commutazione fra console	273
17.3	Prompt e comandi principali	274
17.4	La history	275
17.5	Il server X in più terminali	275
	Scheda n. 26: Inizializzazione e runlevel	276
17.6	L'avvio del sistema	278
17.7	Modificare il manager di avvio	278
17.8	Avviare automaticamente in modalità grafica	279

18 La shell bash	281
18.1 Le shell di Slackware	281
18.2 Standard input e standard output	281
18.3 Comandi di base della bash	282
18.3.1 Elencare file	282
18.3.2 Copiare file e directory	283
18.3.3 Spostare e rinominare file e directory	283
18.3.4 Trovare file	284
18.3.5 Creare un file nuovo	285
18.3.6 Eliminare un file	286
18.3.7 Dividere e ricomporre un file	286
18.3.8 Spostarsi nelle directory	286
18.3.9 Creare una directory	286
18.3.10 Cancellare una directory	287
18.4 L'uso dei caratteri jolly	287
18.5 Comandi intermedi	288
18.5.1 Cercare del testo	288
18.5.2 Le espressioni regolari	288
18.5.3 Vedere del testo	289
18.5.4 Vedere una porzione di testo	289
18.5.5 Ordinare delle righe di testo	290
18.5.6 Contare il testo	290
18.5.7 Concatenare del testo	290
18.5.8 Trovare il formato di un file	290
18.5.9 Registrare una sessione di shell	291
18.5.10 Ottenere informazioni sull'hardware	291
18.5.11 Ottenere informazioni sul sistema	292
18.6 Redirezione dell'input e dell'output	292
18.6.1 Redirezione dello standard output (stdout)	292
18.6.2 Redirezione dello standard input (stdin)	293
18.6.3 Redirezione dell'errore (stderr)	293
18.6.4 Il piping	293
18.7 Ottenere la guida	294
18.7.1 Le pagine man	294
18.7.2 Stampare una pagina man	294
18.7.3 Il comando whatis	294
18.7.4 Il sistema di documentazione info	294
18.7.5 Ricerche con apropos	295
18.8 Personalizzare la shell bash	295
18.8.1 Le impostazioni generali di bash	295
Scheda n. 27: Le shell di Slackware	295
18.8.2 Tipi di shell	296
18.8.3 Personalizzazione di Bash	297
18.8.4 Gli alias dei comandi	298
18.9 Le variabili	299
18.9.1 Tipi di variabili	299
18.9.2 Nome e valore delle variabili	299
18.9.3 Parametri posizionali	300
18.9.4 Variabili incorporate	300
18.9.5 Variabili d'ambiente	300

18.10	Modificare il prompt	300
18.11	Il path	301
18.12	Messaggi di errore	301
18.13	Gli script di shell	302
18.14	Utenti e processi	303
18.14.1	Visualizzare gli utenti e il loro accesso al sistema	303
18.14.2	Visualizzare i processi del sistema	304
18.14.3	Il comando top	306
18.14.4	Gestione dei processi	306
19	Tips & tricks	309
19.1	Installare il kernel 2.6	309
19.1.1	Installare il kernel 2.6 dopo l'installazione	309
19.1.2	Installare il kernel 2.6 al momento dell'installazione	310
19.1.3	Installare un doppio kernel	310
19.2	Velocizzare l'avvio di Slackware	311
19.3	Lanciare programmi all'avvio	312
19.4	Accendere il tastierino numerico all'avvio	312
19.5	Rimuovere fortune	313
19.6	Modificare i caratteri della console	313
19.7	Eliminare il messaggio «mySQL ended»	313
19.8	C'è post@ per te	313
19.9	Regolare la data e l'ora	314
19.10	Tradurre le man pages in italiano	314
19.11	Installare caratteri	315
19.11.1	Il sottosistema font/x	315
19.11.2	Il sottosistema fontconfig	315
19.11.3	L'installazione individuale	315
19.11.4	Aggiungere caratteri con KDE	315
19.12	Stampare con OpenOffice o con StarOffice	316
19.13	Aprire link in Mozilla	316
19.14	Vedere GNU/Linux da Windows	316
19.14.1	Vedere le partizioni EXT2 ed EXT3	317
19.14.2	Vedere le partizioni Reiserfs	317
19.15	Scrivere su NTFS	317
	Ringraziamenti	318
	The GNU General Public License	320
	Abstract	321
	GNU General Public License	321

Introduzione

Premessa

La distro Slackware GNU/Linux è la prima distribuzione uscita fra quelle storiche, conta più anni di vita di ogni altra, e – soprattutto grazie alle innovazioni apportate a partire dalla versione 9.0 – è divenuta maggiormente accessibile agli utenti. Parecchi sono infatti gli strumenti di configurazione (da `xorgconfig` ad `alsaconf`, da `netconfig` a `xwmconfig`, da `liloconfig` a `mouseconfig` e via dicendo) che, sebbene non grafici come quelli di altre distribuzioni, svolgono con molta efficacia il loro compito. Slackware è notoriamente stabile e veloce. Il carattere «spartano» che le viene attribuito (niente strumenti di configurazione di tipo grafico, niente boot splash colorato o con animazione, Lilo senza immagini) può lasciare perplessi utenti che provengono da Microsoft Windows® (ma anche da altre distribuzioni GNU/Linux). Tuttavia ha un suo contraltare per nulla secondario nel fatto che è altamente personalizzabile, in non pochi casi in maniera anche più semplice di altre distribuzioni.

L'utilizzo del kernel vanilla, cioè privo dell'applicazione di patch (a eccezione della patch `speakup`, che introduce funzionalità speciali per l'utilizzo dello schermo per i diversamente abili), ne rende assolutamente praticabile la ricompilazione, che in altre distribuzioni invece è una faccenda assai complicata. È poi facile accorgersi che, in molti casi, proprio questa linearità di Slackware consente di approntare il sistema con pochi ritocchi ai file di configurazione. La filosofia di Slackware è riassunta nel motto KISS, e cioè «Keep It Simple, Stupid»: fin troppo spesso gli strumenti grafici che dovrebbero semplificare la gestione da parte dell'utente complicano il sistema in misura notevole. Inoltre Slackware, spogliata delle GUI (acronimo di Graphical User Interface, e cioè dei frontend grafici), o con un Window Manager che non richiede processori potenti e molta RAM, gira perfettamente anche su vecchi computer dotati di scarse risorse.

Questa guida intende essere un ausilio per ottenere in svariati ambiti una buona configurazione di Slackware in un computer *stand-alone* e una piccola LAN (Local Area Network, nella fattispecie una rete domestica); non prende invece in considerazione l'utilizzo del computer come server web, mail o ftp. Una guida esaustiva è impossibile da scrivere, perché la configurazione varia a seconda dell'hardware e delle esigenze degli utenti. Questo spiega la mole enorme dei quesiti che si leggono nelle mailing list e nei forum, nonché i numerosi How-To, che rimangono essenziali riferimenti per approfondire le questioni e personalizzare ulteriormente il proprio sistema.

Ciò non toglie che sia possibile indicare almeno alcuni elementi di base come punto di partenza della configurazione. Inoltre sono trattati (per lo più nelle schede) alcuni aspetti comuni a tutti i sistemi GNU/Linux, essendo impossibile prescindere. Alcune sezioni sono rivolte a chi voglia installare e configurare Slackware per la prima volta, altre (come la compilazione dei pacchetti, l'uso di `udev` o la condivisione di file e risorse con NFS e Samba) presuppongono invece una qualche confidenza con GNU/Linux e con Slackware. Due sezioni sono dedicate rispettivamente alla console e alla shell `bash`, certo non specifiche di Slackware, ma essenziali per la gestione del sistema.

Nel testo sono state adottate le seguenti convenzioni: le stringhe di testo e le citazioni da sezioni

di file sono in `Courier` su sfondo grigio chiaro; gli indirizzi Internet e i nomi di directory e di file in Helvetica; i comandi in `Courier`; i tasti di comando in **Helvetica grassetto**; le variabili in *corsivo*.

Un cenno storico

Slackware nasce nell'aprile 1993, anche se l'annuncio ufficiale compare in Internet poco tempo dopo, il 16 luglio, e si può ancora leggere all'indirizzo: <http://www.slackware.com/announce/1.0.php>.

Il suo creatore è Patrick J. Volkerding, laureatosi in informatica nel 1993 (il titolo americano è «Bachelor of Science») presso la Moorhead State University del Minnesota. Volkerding, avendo bisogno di un interprete LISP (List Processor, un linguaggio di programmazione di tipo funzionale, notevolmente lontano dai linguaggi imperativi come Fortran, Pascal, C ecc.), costruì un piccolo sistema basato su GNU/Linux, chiamato SLS Linux (Soft Landing System). Fissati alcuni bug di questo sistema, decise di creare un'intera distribuzione, inizialmente riservata a un ristretto gruppo di amici. Slackware 1.0 era composta da due serie di pacchetti: la serie **a** (contenente il kernel, le utilità di base e consistente in tredici floppy) e la serie **x** (contenente il server grafico e consistente in undici floppy). Volkerding continua a essere il mantainer perssoché unico di questa distribuzione, anche se dopo la sua trascorsa malattia aveva dichiarato di voler coinvolgere (ma non si sa a che livello) dei collaboratori, per garantirle un futuro in modo del tutto indipendente dalle sue vicende personali. Inoltre, si affida a un gruppo ristretto di persone che seguono gli aggiornamenti di sicurezza. Si tratta insomma, per Slackware, di una «one man distro».

Le varie release si susseguono con scadenza approssimativamente annuale, ma senza autentica regolarità (il salto nella numerazione dalla 4 alla 7 è stato determinato da motivazioni commerciali e di immagine), con in più alcune versioni intermedie per aggiornamenti e bug fixes:

1.0	17/07/1993	3.4	14/10/1997
1.0.1	04/08/1993	3.5	09/06/1998
1.0.2	05/09/1993	3.6	28/10/1998
1.0.3	15/09/1993	3.9	10/05/1999
1.0.4	01/10/1993	4.0	17/05/1999
1.1.0	04/11/1993	7.0	25/10/1999
1.1.2	27/03/1994	7.1	22/06/2000
2.0	25/06/1994	8.0	28/06/2001
2.1	15/11/1994	8.1	19/06/2002
2.2.0	30/03/1995	9.0	20/03/2003
2.3	24/05/1995	9.1	25/09/2003
3.0	30/09/1995	10	22/06/2004
3.1	03/07/1996	10.1	06/02/2005
3.2	17/02/1997	10.2	14/09/2005
3.3	11/07/1997	11.0	02/10/2006

I curiosi o i nostalgici possono tuttora scaricare (e provare a installare) le vecchie versioni di Slackware all'indirizzo: <http://zuul.slackware.biz/versions>.

Il nome di questa distribuzione GNU/Linux deriva dal termine «slack», che come sostantivo significa «ozio», «pigrizia», «indolenza», e può essere usato anche come aggettivo, nelle medesime accezioni. Dietro questa curiosa denominazione sta l'adesione alla *Church of the SubGenius*, movimento pseudo-religioso «tecno-pagano» nato a Dallas nel 1953, che acquistò popolarità durante gli anni Ottanta e Novanta. Esso presentava evidenti legami con la cultura pop, con la musica underground e uno spiccato interesse per Internet. Attualmente, il sito di riferimento del movimento è: <http://www.subgenius.com>.

La fondamentale credenza della Chiesa dei Subgenius è la ricerca dello Slack, ovvero un senso di libertà, di indipendenza, di autonomia allorché si riescono a raggiungere i propri obiettivi. Tutti nascono con lo spirito dello Slack, ma poi esso viene inibito dalla «cospirazione» della gente comune e dal conformismo. Di conseguenza, i seguaci dello Slack non devono accettare l'autorità costituita e i limiti della società.

Requisiti

Slackware è una distribuzione a 32 bit per computer PC compatibili con processore Intel 486 o superiore (Intel, Celeron, Pentium, MMX, AMD, ecc.); tuttavia ne esistono oggi dei port per altre architetture: uno ufficiale per s/390 e altri non ufficiali: Slackintosh per powerpc (<http://workaround.ch/index.html>) e Splack per sparc (<http://www.splack.org>). Inoltre, secondo i dati che è possibile reperire in siti quali <http://www.distrowatch.com> e <http://www.linux.org>, esistono attualmente quasi quaranta cloni di Slackware e una ventina di distribuzioni live basate su di essa (la più nota è probabilmente Slax, reperibile all'indirizzo: <http://slax.linux-live.org>). Slackware attualmente è ottimizzata per i686, anche se è compatibile con processori da i486 in su. Inoltre contiene un completo ambiente di sviluppo che supporta C, C++, Fortran-77, LISP, Java e altri linguaggi.

Slackware richiede almeno 16 MB di RAM, poco più di 3 GB di spazio sul disco fisso per un'installazione completa, e necessita di un file di swap (sebbene sarebbe possibile farne a meno in presenza di una accertata RAM sufficiente). Si possono installare soltanto i programmi inclusi nella serie di pacchetti **a** (che costituisce la configurazione minimale e consente comunque al sistema di funzionare), aggiungendo in un secondo momento altri pacchetti. Molti sono i lettori CD anche obsoleti supportati (in certi casi, lettori apparentemente non supportati usano in realtà componenti elettronici di ditte supportate). In questo caso si può andare per tentativi. I lettori con standard IDE/ATAPI sono tutti supportati, mentre per quelli SCSI tutto dipende dal supporto al controller.

Informazioni su Slackware

La prima fonte di informazioni su Slackware (e su GNU/Linux in generale) si trova nella distribuzione stessa, ricca di documentazione che si può reperire in `/usr/doc/Linux-HOWTOs` ed eventualmente scaricabile anche all'indirizzo del Linux Documentation Project: <http://www.tldp.org>, parzialmente mirrorato in lingua italiana all'indirizzo: <http://it.tldp.org>.

Qui si possono vedere gli How-To, le FAQ, gli LDP e infine le LDPInstall-guide. Sebbene in alcuni casi questi documenti siano un po' datati, costituiscono comunque fonti preziose di informazione.

La guida ufficiale, ampia ma non onnicomprensiva, è lo *Slackware Linux Essentials*, di cui è stata rilasciata la seconda edizione, contenuta nel DVD o nel CD 5 nell'omonima directory della distribuzione, che si può inoltre scaricare in rete all'indirizzo: <ftp://ftp.slackbook.org/pub/slackbook/slackbook-2.0.pdf>, oppure acquistare in formato cartaceo.

Una versione italiana della prima edizione è presente all'indirizzo: <http://www.slackit.org/book.php>. Una traduzione della seconda edizione è attualmente in corso.

Capitolo 1

Procurarsi Slackware

1.1 Scaricare Slackware

Il primo sito in cui è possibile scaricare le ISO di Slackware (ora non soltanto dei CD, ma anche del DVD) è, naturalmente, quello stesso della distribuzione, all'indirizzo: <http://www.slackware.com>. Ci sono poi svariati mirror, segnalati all'indirizzo: <http://www.slackware.com/getslack> e altri facilmente individuabili mediante i motori di ricerca come Google. Infine, è possibile scaricarle mediante BitTorrent (vedi Scheda n. 1 – Condivisioni con BitTorrent). Un lungo elenco di mirror http, ftp e rsync si legge all'indirizzo: <http://alphageek.dyndns.org/linux/slackware-mirrors.shtml>.

Come quasi tutte le distribuzioni GNU/Linux, anche Slackware si trova in allegato a parecchie riviste. Se si intende utilizzarla stabilmente, non è inopportuno acquistarla o abbonarsi (per ricevere gli aggiornamenti), così da fornire un supporto economico anche se minimo agli sviluppatori. Il set consta di sei CD o di un DVD perché contiene tutti i sorgenti.

Scheda n. 1

Condivisioni con BitTorrent

Fino a un paio di anni fa, in Internet dominava una netta distinzione fra i server e i client. I server fornivano contenuti che potevano essere scaricati dai client. Ciò era dovuto anche ai limiti oggettivi della banda: chi stava scaricando con un modem a 56KB non aveva banda sufficiente per condividere file. Con l'avvento delle tecnologie broadband come la DSL si è resa disponibile una banda notevolmente più ampia, che ha portato alla nascita dei p2p, acronimo di «peer-to-peer», che letteralmente significa «da pari a pari» (per sottolineare il fatto che tutti i nodi rivestono, indistintamente, lo stesso ruolo). Fra questi, una soluzione particolarmente felice è costituita da BitTorrent, creato da Bram Cohen e reperibile all'indirizzo: <http://www.bittorrent.com>.

La «filosofia» di BitTorrent

La «filosofia» del p2p risiede nel fatto che chi è connesso per scaricare qualcosa può rendere disponibile quello che ha già scaricato, o altro materiale che risiede nel suo computer: dunque ogni client si trasforma in un piccolo server. Mentre però in alcuni sistemi di p2p la ricerca dei file è integrata nel programma stesso, BitTorrent si limita al download, lasciando all'utente il compito di trovare la fonte (ad esempio, mediante un motore di ricerca). Nel sito di Slackware sono indicati molti BitTorrent. In secondo luogo, innumerevoli programmi di p2p consentono di scaricare file senza preoccuparsi se l'utente consenta la condivisione dei propri. Ma in questo modo sarebbe assai difficile farli funzionare adeguatamente: scaricare ciò che ci interessa da altri senza consentire che, a loro volta, altri possano scaricare da

noi, viene chiamato *leeching* («fare la sanguisuga»). Per evitare questo problema BitTorrent impone una propria tecnica: suddivide il file richiesto in piccoli pezzi (detti *chunk*, di 250 KB l'uno), velocizzando il download e rendendolo più efficiente, perché è possibile scaricare anche da chi non ha il file per intero, ma lo sta scaricando contemporaneamente a noi. Inoltre, per combattere il *leeching*, BitTorrent permette di iniziare a scaricare un file anche a chi non ne ha messo a disposizione nessuno, ma se poi nulla viene messo a disposizione, non fa più proseguire l'operazione.

Funzionamento di BitTorrent

Per poter usufruire del sistema è necessario, innanzitutto, prelevare da un sito web dedicato un file di estensione *.torrent*. Si tratta di un file statico, contenente delle informazioni codificate mediante un algoritmo di *hashing*: il file da prelevare e/o da trasferire, l'indirizzo di un server traccia (in inglese *tracker*) utilizzato per localizzare i sorgenti (cioè altri utenti) che possiedono il file o parte di esso e altre notizie e statistiche. Il client BitTorrent si connette al server traccia e ottiene una lista degli utenti che hanno scaricato o stanno scaricando lo stesso torrent. Allo stesso tempo, esso aggiunge il nuovo utente alla lista, di modo che altri a loro volta possano scaricare da quest'ultimo. Il client aggiorna periodicamente la lista degli utenti.

In questo modo, ogni nodo contribuisce a diffondere il file: più è ampia la banda su un determinato nodo, più elevate sono le probabilità che il *tracker* venga diffuso verso altri nodi. Ciò è favorito dal funzionamento stesso di BitTorrent: come abbiamo visto, quando parte una richiesta di download, BitTorrent inizia il prelievo solo se c'è un numero sufficiente di utenti connessi (almeno uno dei quali deve possedere il file completo), laddove, se ce ne sono pochi, è possibile condividere solo alcune parti del file.

La velocità di scaricamento viene così progressivamente incrementata quanto più numerosi sono gli utenti che stanno scaricando il file. Inoltre, i server traccia che possiedono il file completo, e che vengono chiamati semi (in inglese *seed*), fanno sì che nel computer dell'utente lo scaricamento di un determinato file si completi solo nel momento in cui essi hanno completato la copia del file stesso verso altri server traccia, per consentirne la diffusione.

BitTorrent e i firewall

Se il traffico è controllato da un firewall, per consentire sia il download sia l'upload va abilitato il traffico in TCP e UDP per tutte le porte di cui si serve BitTorrent, e precisamente le porte 6881-6889. Se il firewall facesse il NAT (Network Address Translation), i pacchetti non potrebbero raggiungere l'indirizzo privato del client e ciò renderebbe impossibile la condivisione. È pertanto necessario configurare adeguatamente il router, a seconda del modello. IPTables richiede invece una regola del tipo:
`iptables -t nat -I PREROUTING -p tcp --dport 6881:6889 -j DNAT --to-destination host`
 dove *host* è l'indirizzo IP privato su cui gira il client BitTorrent.

Se invece non si usa un router e la connessione a Internet non è stabile (quando ci si connette con un modem, l'indirizzo IP varia sempre; con una connessione tipo DSL ciò è meno probabile, ma mai garantito), è necessario comunicare al server traccia il corretto indirizzo IP pubblico con l'opzione `--ip 123.45.67.890`. In questo caso si deve aggiornare la riga di comando, che risulterà del tipo:
`bittorrent --ip 123.45.67.890 --responsefile "%1`

In alternativa, un sistema più farraginoso è ricorrere a servizi esterni (come i siti <http://checkip.dyndns.org>, <http://www.whatismyipaddress.com>, <http://www.dnsart.com>).

I comandi principali di BitTorrent

BitTorrent funziona da console col comando: `bittorrent-console file.torrent`, con la versione ncurses, sempre da console, col comando: `bittorrent-curses file.torrent` e si vale invece di un'interfaccia grafica (versione GUI) col comando: `bittorrent file.torrent`. Quest'ultima versione richiede che siano installate le librerie wxpython (che di default non sono presenti in Slackware).

In `/etc/mailcap` dovrà essere presente la riga:

```
application/x-bittorrent; /usr/bin/bittorrent-xterm '%s';
```

che solitamente viene aggiunta in modo automatico all'installazione del pacchetto. I passaggi pratici da seguire sono allora i seguenti:

- a) individuare il file che ci interessa (ad esempio, la ISO n. 1 di Slackware)
- b) scaricare il relativo file `.torrent` (nel nostro caso quello relativo alla ISO n. 1 di Slackware, reperibile all'indirizzo: <http://www.slackware.com/getslack/torrents.php>)
- c) lanciare da console il comando a seconda dell'interfaccia che desideriamo, ad esempio:
`bittorrent-curses slackware-11.0-install-d1.torrent`

Per scaricare (o diffondere) più file senza lanciare molteplici istanze del programma, si può usare la versione del client `launchmany-console` oppure `launchmany-curses`. Il primo parametro sulla linea di comando specificherà una cartella, che contiene un certo numero di `.torrent` e i loro corrispondenti file/cartelle; il resto della riga di comando segue le normali regole di BitTorrent.

Con i comandi `maketorrent` (da GUI) e `maketorrent-console` (da console) è semplicissimo creare un file torrent per i file o le directory che si vogliono mettere a disposizione in questo modo.

Slackware presenta BitTorrent nella sezione `/extra` (CD 2), e inoltre BitTornado, una versione sperimentale modificata di BitTorrent (che richiede anch'esso le librerie `wxpython`).

BitTorrent e i browser

Per automatizzare lo scaricamento dal browser, è necessario creare un'associazione tra i file di estensione `.torrent` e `/usr/bin/btdownloadxterm.sh`. Una volta scaricato il torrent, si aprirà così automaticamente la finestra di scaricamento. Infine, vi sono alcune GUI molto buone per BitTorrent: Azureus, scaricabile dal sito: <http://azureus.sourceforge.net> (che richiede la presenza di Java) implementa per l'appunto il protocollo BitTorrent.

1.2 Controllare le ISO

1.2.1 Quali controlli

Nel caso si scarichino le ISO da un sito web (BitTorrent verifica automaticamente la correttezza dei file scaricati), esiste un modo per controllare se esse sono corrotte oppure no. Ogni ISO infatti può essere accompagnata (e in genere lo è) da un file di ridotte dimensioni e di estensione `.md5` o anche da un file di estensione `.asc`. Nel primo caso si tratta del digest della ISO stessa, ossia di una sorta di impronta digitale generata mediante un algoritmo che «riassume» il contenuto della ISO, nel secondo caso della firma digitale della ISO.

1.2.2 Lo md5sum

Per controllare la ISO si deve lanciare il comando: `md5sum nome_iso.iso` e confrontare l'output con la stringa contenuta nel file di digest che la accompagna. Il comando sarà dunque, ad esempio: `md5sum slackware-11.0-install-d1.iso`. La stringa risultante dovrà essere identica a quella presente nel file `slackware-11.0-install-d1.iso.md5`.

1.2.3 La firma digitale

Il file `.asc` è la firma GPG applicata al file e serve a garantirne in aggiunta l'autenticità (e perciò il fatto che nessuno lo ha modificato). Infatti un malintenzionato potrebbe modificare il file originale e produrre un nuovo `md5sum`; in questo modo invece abbiamo la garanzia che il file proviene davvero da una fonte attendibile. Ciò si rivela tanto più utile se il file che riceviamo non proviene da un sito ufficiale, ma da fonti anonime e di per sé incontrollabili, come quando si usano programmi di file sharing. Per

fare questo si ricorre a GnuPG, un programma che gestisce la crittografia a chiave pubblica delle e-mail e le firme digitali (vedi [Scheda n. 2 — GnuPG: firma e crittografia](#)).

Per verificare le ISO di Slackware, dobbiamo compiere i passaggi seguenti:

a) ci procuriamo la firma GPG di Slackware (molti siti hanno una sezione sicurezza con tali firme), reperibile all'indirizzo: <http://www.slackware.com/gpg-key>.

b) Salviamo la firma in un file di solo testo, con un nome a piacere ed estensione .asc (ad esempio slackware.asc).

c) Importiamo da console la firma nel nostro database di firme (il cosiddetto *keyring*) col comando:

```
gpg --import chiave.asc (ad esempio gpg --import slackware.asc).
```

d) Possiamo eventualmente controllare le chiavi già importate col comando: `gpg --list-keys` e dovrà apparirci anche la chiave appena importata.

e) Firmiamo la chiave con la nostra tramite il comando: `gpg --edit-key chiave` (ad esempio `gpg --edit-key slackware`) ed entriamo quindi in modalità comando. Diamo quindi il comando: `sign`.

Di seguito la chiave dev'essere anche convalidata. Se siamo certi dell'autenticità della firma, le attribuiamo un determinato livello di affidabilità che va da nessuna convalida a convalida totale e definitiva. Confermiamo la scelta ed effettuiamo l'operazione con la nostra firma GPG.

A questo punto abbiamo installato la firma dell'autore; siamo pronti adesso a verificare la ISO. Scarichiamo anche il file di firma di quest'ultima e diamo il comando: `gpg --verify nome_iso.asc nome_iso.iso`. Il risultato ci dirà se la firma è valida.

1.2.4 Controllo dei file

È sempre opportuno controllare lo md5sum di ogni file scaricato, non solo delle ISO, per accertarsi della correttezza del file. È possibile anche controllare sequenzialmente una serie di pacchetti coi relativi md5sum col comando: `md5sum -c checksums.md5 | less`

Scheda n. 2

GnuPG: firma e crittografia

Il programma GnuPG è la versione open source di PGP, e cioè Pretty Good Privacy. Si tratta di un sistema di autenticazione e di crittazione a chiave pubblica che rende possibile inviare mail firmate o crittate in modo che solo il destinatario possa leggerle o che possa comunque verificare l'autenticità del mittente e che, allo stesso modo, rende possibile apporre a qualsiasi file una firma digitale per consentire il controllo della sua autenticità. Per svolgere tutte queste operazioni vengono usate da ciascun utente due chiavi, una privata e una pubblica, generate mediante appositi comandi. La chiave privata dev'essere tenuta segreta e serve per decrittare i messaggi ricevuti, quella pubblica invece dev'essere inviata ai destinatari. Quando A deve mandare un messaggio crittato a B, lo critta con la chiave pubblica di B, il quale poi lo decritta con la sua chiave privata. Invece, per firmare A applicherà al messaggio la sua chiave privata e B potrà verificare mediante la chiave pubblica di A che il messaggio è stato in effetti inviato da questi.

Generare le proprie chiavi

Per poter usare GnuPG, la prima tappa consiste nel generare la propria coppia di chiavi, pubblica e privata, che saranno collocate nella directory nascosta `/.gnupg` della home. A questo fine si usa il comando: `gpg --gen-key`. Di norma, si usa il tipo proposto di default e, come lunghezza del file, 1024 bit (una maggiore lunghezza aumenta la sicurezza e insieme i tempi di crittazione e decrittazione). L'algoritmo maggiormente adottato è DSA/ElGamal, in quanto non brevettato. Durante la generazione della firma è opportuno aumentare l'entropia (ossia il «disordine») del computer, muovendo il mouse e aprendo altri programmi.

Tuttavia questa chiave è relativamente lunga (anche se si adotta la versione a 1024 bit) e quindi scomoda da inserire. Per ovviare a questo inconveniente, spesso si usa firmare i messaggi con una versione della chiave non in formato binario, ma ascii (detta *ascii-armored*, un formato protetto da un'armatura ascii simile ai documenti codificati con uuencode), che è più breve. I file di chiave binari di solito hanno l'estensione *.gpg* (o *.sig*), quelli ascii l'estensione *.asc*. Per creare questo secondo tipo di chiave e visualizzarla si adotta l'opzione `--armor` (anche abbreviata in `-a`) e serve il comando: `gpg --armor --export -o firma.asc`, dove l'opzione `-o` abbrevia `--output` e fa vedere la firma sullo standard output (lo schermo).

Per usare la chiave si deve creare anche una password (detta *passphrase*), per la quale valgono le stesse regole di opportunità di quando si crea una password per altro uso.

Per cancellare la chiave pubblica si usa il comando: `gpg --delete-key ID`.

Per cancellare la chiave privata si usa il comando: `gpg --delete-secret-key`.

Per generare un certificato di revoca, atto a revocare la nostra chiave se per qualche motivo intendessimo eliminarla, si usa il comando: `gpg --gen-revoke nome_utente`. Infatti non potremo, nell'eventualità, ricreare la chiave originaria, in quanto anche fornendo gli stessi dati (e-mail, passphrase ecc.) la firma risultante sarebbe comunque diversa.

Esportare la propria chiave pubblica

La chiave pubblica può essere inviata manualmente ai nostri corrispondenti, oppure può essere pubblicata in appositi server. Essa può essere quindi inclusa o allegata ai messaggi.

L'esportazione si effettua, se la chiave binaria, col comando: `gpg --export ID chiave.gpg`, mentre se è ascii col comando: `gpg -a --export ID chiave.asc`. L'esportazione in un keyserver si effettua col comando: `gpg --keyserver keyserver --send-key ID`. Altro semplicissimo metodo consiste nell'effettuare un copia-incolla dell'output di: `gpg -a --export ID` nell'apposito form della pagina web del server delle chiavi.

Importare e convalidare le chiavi

Per importare le chiavi pubbliche dei corrispondenti nel database delle nostre chiavi (detto *keyring*) si usa il comando: `gpg --import chiave.asc`, mentre per verificare se l'importazione ha avuto effettivamente luogo, o semplicemente per controllare quali chiavi sono state importate, si dà il comando: `gpg --list-keys`.

Ogni chiave appena importata dev'essere convalidata firmandola con la nostra chiave. A questo fine si usa il comando: `gpg --edit` che ci fa entrare in modalità comando, e quindi il comando: `sign`, dopo di che si attribuisce alla chiave un determinato livello di affidabilità, sempre da modalità comando, col comando: `trust` e si sceglie il livello di affidabilità da conferire alla chiave importata e appena firmata.

Verificare una firma

Per verificare una firma si usa il comando: `gpg --verify chiave.asc file_da_verificare`.

Firmare e crittare un file

Per firmare un file si dà il comando: `gpg --sign file`.

Per firmarlo con una firma ascii si dà il comando: `gpg -a --clearsign file`.

Per firmare un file con una forma ascii separata, che può consentire di verificare il file stesso, si usa il comando: `gpg -sba file`. Questo comando produce un file separato di nome *file.asc*, di cui ci si serve per la verifica. Le opzioni stanno rispettivamente per: `-s` = firmare il file; `-b` = produrre una firma separata (detta *detach sign*); `-a` = produrre una firma in formato ascii (come abbiamo già visto).

Per crittare un file si dà il comando: `gpg -e file`, dove l'opzione `-e` abbrevia `--encrypt`, oppure, sempre con firma ascii: `gpg -a -e file`.

Per crittare un file con un nome determinato, il comando sarà: `gpg --output file_crittato --encrypt --recipient ID file_da_crittare`.

Decrittare un file

Per decrittare un file si usa il comando: `gpg [-d] file`, dove l'opzione `-d` (facoltativa) abbrevia `--decrypt`, mentre per decrittare e controllare un file si usa il comando: `gpg -d chiave.asc file`.

Per decrittare un file con un nome determinato, il comando sarà: `gpg --output file_crittato --decrypt nome_file_decrittato`.

L'output predefinito può essere rediretto a un file con l'opzione `-o file`.

I client di posta

La maggior parte dei client di posta, sia testuali (come Mutt) sia grafici (come Mozilla Thunderbird, che ricorre a un add-on quale Enigmail), riescono a gestire le firme GnuPG. Kmail presenta un'ottima integrazione con GnuPG e, se si sceglie di firmare le e-mail, usa di default la firma *ascii-armored*. Inoltre KDE ha un semplice strumento grafico (KGpg) per svolgere tutte le operazioni che sono state descritte. Le opzioni di GnuPG sono molte di più di quelle segnalate in questa scheda. Un'ampia documentazione è reperibile nel sito del programma, all'indirizzo: <http://www.gnupg.org>, mentre per le opzioni a riga di comando si può ricorrere alla pagina man di `gpg`.

1.3 Masterizzare le ISO

1.3.1 Masterizzare le ISO su CD col kernel 2.4

Per masterizzare le ISO di CD da GNU/Linux, si può ricorrere al programma `cdrecord` e alla linea di comando. Poniamo qui di usare un masterizzatore ATAPI. Se si usa un kernel della serie 2.4, va abilitata l'emulazione SCSI (anche se le versioni più recenti di `cdrecord` non lo richiedono). In questo caso, dapprima è necessario individuare l'indirizzo del masterizzatore, costituito dalla tripla di bus, id e lun (il cosiddetto Logical Unit Number del dispositivo), col comando: `cdrecord --scanbus` che fornisce una tripla come 0,1,0 ecc. A questo punto si può passare direttamente alla masterizzazione della ISO su CD col comando: `cdrecord dev=/0,1,0 -v nome_iso.iso`

1.3.2 Masterizzare le ISO su CD col kernel 2.6

Se si usa un kernel della serie 2.6, che non richiede più l'emulazione SCSI (la quale anzi è ormai deprecata), l'indirizzo del masterizzatore va trovato col comando: `cdrecord --dev=ATAPI --scanbus`. Ipotizziamo che tale tripla sia data dalla stringa 0,1,0.

Una volta individuata la tripla, si passa alla masterizzazione della ISO su CD col comando: `cdrecord dev=ATAPI:0,1,0 -v [opzioni] nome_iso.iso` oppure, più semplicemente, col comando: `cdrecord dev=/dev/hdn -v [opzioni] nome_iso.iso`, dove *n* è il numero del dispositivo.

1.3.3 Controllo del CD

Per verificare la corrispondenza fra la ISO e il CD appena creato, in GNU/Linux si può ricorrere al comando: `cmp /dev/cdrom nome_iso.iso`, oppure rigenerare lo md5sum dal CD col comando: `md5sum /dev/cdrom`.

1.3.4 Creare ISO da CD

Aggiungiamo incidentalmente che, all'occorrenza, si può anche creare una ISO da CD col comando: `dd if=/dev/cdrom of=nome_iso.iso`.

1.3.5 Reperire Slackware in DVD

Benché si tratti di realizzazioni non ufficiali, le varie versioni di Slackware anteriori alla 11.0 sono reperibili anche in ISO per DVD all'indirizzo: <ftp://ftp.slackware.no/pub/linux/ISO-images/Slackware>.

1.3.6 Masterizzare le ISO su DVD

Per masterizzare le ISO di DVD da GNU/Linux, si può ricorrere al programma `dvd+rw-tools` (non prendiamo qui in considerazione l'alternativa costituita da `cdrecord` e `provd`, programmi peraltro non presenti in Slackware) e alla linea di comando. È sufficiente dare da console: `growisofs -dvd-compact -Z /dev/dvd=nome.iso`.

In questo contesto, l'opzione `-Z` serve a supportare la scrittura di immagini ISO già pronte e l'opzione `-dvd-compact` garantisce la massima compatibilità con i DVD-ROM, producendo un DVD in sessione unica (cioè chiuso).

Si possono aggiungere altre opzioni: `-R` per abilitare l'estensione RockRidge, `-J` per abilitare l'estensione Joliet (atta a far leggere correttamente il DVD sotto Windows), `-speed=n` per ricorrere a una determinata velocità (*n*) di masterizzazione,

1.3.7 I front-end grafici

In alternativa, si possono usare i front end grafici, fra i quali attualmente il più diffuso e potente è **k3b** (contenuto negli `/extra` del CD 3 di Slackware, dove peraltro è presente anche un'altra GUI per masterizzare, e cioè **xcdroast**).

In questo caso, si devono scegliere rispettivamente le seguenti voci di menu: **Strumenti – CD – Scrivi immagine CD** oppure **Strumenti – DVD – Scrivi immagine ISO DVD** o ancora cliccare semplicemente col mouse sul nome della ISO e si aprirà lo strumento adatto. Allo stesso modo, è possibile gestire questa fase da Windows (ad esempio con Ahead Nero®). In tutti i casi, non bisogna dimenticare che per masterizzare le ISO esiste un metodo apposito, diverso dalla masterizzazione di semplici CD di dati (in generale, va cercato un comando simile a «scrivi immagine»).

Capitolo 2

Preparare il disco fisso

2.1 Partizionare il disco fisso

2.1.1 Partizioni richieste

Attualmente, un disco fisso non può contenere più di quattro partizioni primarie; per averne in numero maggiore, è necessario creare una partizione estesa che, a sua volta, può contenere fino a dodici partizioni logiche.

L'esigenza minimale di Slackware è avere due partizioni, una per l'installazione del programma e una eventuale per il file di swap (utile a reperire memoria nel caso sia insufficiente quella fornita dalla RAM del computer); sarà questo lo scenario che considereremo. Naturalmente questa configurazione può essere modificata a seconda delle esigenze di ciascuno. Ad esempio, si può dedicare una partizione diversa atta a contenere le cartelle home degli utenti (vedi [Scheda n. 4 – La partizione /home](#)) oppure la directory `/etc`, che contiene tutti i file comuni di configurazione. C'è chi crea ulteriori partizioni per `/var` o per `/tmp`. Tutto dipende dalle esigenze soggettive, ma noi non considereremo questi casi più complessi.

Nel caso si debba reinstallare il sistema, la suddivisione del disco fisso in svariate partizioni consente di non perdere impostazioni e dati. È anche possibile usare un secondo disco fisso per archiviare i dati: in questo ambito esiste la massima elasticità e nessuna regola fissa. Non dimentichiamo che tutte le unità, siano esse dischi o partizioni, sono viste da GNU/Linux come dei file; ciò rende molto semplice creare e gestire una serie di partizioni differenti in più dischi fissi, come pure effettuare operazioni quali l'aumento, il ridimensionamento o lo spostamento delle partizioni stesse.

2.1.2 Due casi esemplificativi

Partiamo da due ipotesi: 1) un disco fisso che contiene già Microsoft Windows (poiché, nella maggior parte dei casi, lo abbiamo già come sistema operativo); 2) un disco fisso nuovo, si suppone privo di qualsiasi formattazione.

Nel primo caso è necessario creare spazio per Slackware. Innanzitutto bisogna deframmentare il disco fisso (il problema della frammentazione è ben noto agli utenti di Windows, mentre non tocca quelli di GNU/Linux, grazie alla maggiore efficienza dei suoi file system nativi, in virtù dei quali la deframmentazione si assesta intorno al 4-5%) e quindi creare le partizioni per Slackware. Anche qui ci si presentano svariate alternative. Operiamo con uno strumento di gestione delle partizioni per Windows come Partition Magic[®] della PowerQuest (esso consente anche di creare due floppy avviabili con cui si partiziona il disco fisso in modo molto user-friendly in modalità pseudo-grafica, con l'aiuto del mouse, senza avviare Windows). Esiste uno strumento open source come Parted (presente nella directory `/extra`

del CD 3 di Slackware), il quale però supporta NTFS solo se opportunamente patchato; ma se supponiamo di avere solo Windows nel nostro computer per il momento non saremo in grado di utilizzarlo. Peraltro, quest'ultimo programma, come altri per GNU/Linux, è in grado di gestire un elevato numero di file system, ben più vario di quelli accessibili dagli strumenti di amministrazione per Windows.

Utilizzeremo allora i floppy per ridimensionare la partizione di Windows (sia essa FAT32 o NTFS) e creeremo altre due partizioni, una EXT3 (Partition Magic consente di creare partizioni GNU/Linux di due formati, EXT2 ed EXT3, più il file system per il file di swap, mentre non ne supporta altri, come Reiserfs) e una di swap. La prima dev'essere sufficientemente ampia da ospitare il sistema operativo (nel caso di un'installazione completa, poco più di 3 GB) e i programmi aggiuntivi; la seconda, alla luce di un antico adagio, dovrebbe ammontare al doppio della RAM disponibile. Tuttavia ciò è valido solo se si ha poca RAM; oggi, non essendo infrequente trovare computer domestici che hanno un GB o più di RAM, si può creare un file di swap più piccolo o addirittura ometterlo in presenza di un'ampia RAM. Se poi vogliamo, creeremo anche una partizione /home per gli utenti (ciò vale – ed è decisamente consigliabile – anche nel caso che siamo noi l'unico utente del nostro computer). La sua dimensione dipende ovviamente da che cosa essa è destinata a contenere.

2.2 Usare Cfdisk

2.2.1 Fdisk e Cfdisk

Se vogliamo, possiamo creare la partizione direttamente, al momento stesso dell'installazione, con gli strumenti che Slackware ci mette a disposizione, sia se abbiamo nel disco fisso un preesistente sistema operativo, sia se il nostro disco fisso è ancora intonso, come da seconda ipotesi. Facciamo ricorso a `cfdisk`, che ha un'interfaccia testuale un po' più amichevole di `fdisk`. L'unico problema è che queste due utilità, se si modificano partizioni esistenti, ne eliminano il contenuto: vale sempre l'aurea regola di fare un backup prima di qualsiasi operazione di ridimensionamento.

2.2.2 Creare le partizioni

In quest'ultimo caso dobbiamo preparare il disco con due passaggi. Innanzitutto creiamo le partizioni per i vari file system che intendiamo adottare (il che corrisponde alla formattazione a basso livello di Windows: tali partizioni saranno pertanto ancora prive di un file system). Lanciato il programma, se il disco fisso è vuoto il programma ci chiede se intendiamo attivarlo da «zero table»; rispondiamo affermativamente «y»:



Dal menu in basso, scegliamo «New» (= «nuova partizione»):

```

cfdisk 2.12
Disk Drive: /dev/hda
Size: 10737418240 bytes, 10.7 GB
Heads: 255 Sectors per Track: 63 Cylinders: 1305
-----
Name      Flags      Part Type  FS Type    [Label]    Size (MB)
-----
Pri/Log   Free Space 10734.00

[ Help ] [ New ] [ Print ] [ Quit ] [ Units ]
[ Write ]

Create new partition from free space_

```

Quindi «Primary» (= «primaria»):

```

cfdisk 2.12
Disk Drive: /dev/hda
Size: 10737418240 bytes, 10.7 GB
Heads: 255 Sectors per Track: 63 Cylinders: 1305
-----
Name      Flags      Part Type  FS Type    [Label]    Size (MB)
-----
Pri/Log   Free Space 10734.00

[Primary] [Logical] [Cancel ]

Create a new primary partition_

```

Digitiamone la dimensione in MB, ricordando che 1 GB = 1024 MB, o 1000 MB per semplificare il computo (nel nostro esempio, 5000 MB):

```

cfdisk 2.12
Disk Drive: /dev/hda
Size: 10737418240 bytes, 10.7 GB
Heads: 255 Sectors per Track: 63 Cylinders: 1305
-----
Name      Flags      Part Type  FS Type    [Label]    Size (MB)
-----
Pri/Log   Free Space 10734.00

Size (in MB): 5000_

```

Scegliamo di collocarla all'inizio dello spazio libero presente nel disco fisso (presumibilmente dopo la partizione che ospita Windows):

```

cfdisk 2.12

Disk Drive: /dev/hda
Size: 10737418240 bytes, 10.7 GB
Heads: 255 Sectors per Track: 63 Cylinders: 1305

-----
Name      Flags      Part Type  FS Type    [Label]    Size (MB)
-----
          Pri/Log   Free Space 10734.00

[Beginning] [ End ] [ Cancel ]

Add partition at beginning of free space_

```

Andiamo nel menu che ci permette di scegliere il tipo di partizione che vogliamo creare:

```

cfdisk 2.12

Disk Drive: /dev/hda
Size: 8589934592 bytes, 8589 MB
Heads: 255 Sectors per Track: 63 Cylinders: 1044

-----
Name      Flags      Part Type  FS Type    [Label]    Size (MB)
-----
hda1     Primary   Linux      5000.98
          Pri/Log   Free Space 3586.23

[Bootable] [ Delete ] [ Help ] [Maximize] [ Print ]
[ Quit ] [ Type ] [ Units ] [ Write ]

Change the filesystem type (DOS, Linux, OS/2 and so on)_

```

Infine scegliamo il tipo di formattazione fra le numerose possibilità che il menu ci offre:

```

01 FAT12          4E QNX4.x 2nd part    A7 NeXTSTEP
02 XENIX root    4F QNX4.x 3rd part    A8 Darwin UFS
03 XENIX usr     50 OnTrack DM         A9 NetBSD
04 FAT16 <32M   51 OnTrack DM6 Aux1   AB Darwin boot
05 Extended     52 CP/M              B7 BSDI fs
06 FAT16        53 OnTrack DM6 Aux3   B8 BSDI swap
07 HPFS/NTFS    54 OnTrackDM6        BB Boot Wizard hidden
08 AIX          55 EZ-Drive          BE Solaris boot
09 AIX bootable 56 Golden Bow       C1 DRDOS/sec (FAT-12)
0A OS/2 Boot Manager 5C Priam Edisk      C4 DRDOS/sec (FAT-16 <
0B W95 FAT32    61 SpeedStor        C6 DRDOS/sec (FAT-16)
0C W95 FAT32 (LBA) 63 GNU HURD or SysU  C7 Syrinx
0E W95 FAT16 (LBA) 64 Novell Netware 286 DA Non-FS data
0F W95 Ext'd (LBA) 65 Novell Netware 386 DB CP/M / CTOS / ...
10 OPUS        70 DiskSecure Multi-Boo DE Dell Utility
11 Hidden FAT12  75 PC/IX            DF BootIt
12 Compaq diagnostics 80 Old Minix       E1 DOS access
14 Hidden FAT16 <32M 81 Minix / old Linux E3 DOS R/O
16 Hidden FAT16  82 Linux swap       E4 SpeedStor

Press a key to continue_

```

Per GNU/Linux si tratta del file system 83, per il file di swap del file system 82:

```

17 Hidden HPFS/NTFS      83 Linux          EB BeOS fs
18 AST SmartSleep       84 OS/2 hidden C: drive EE EFI GPT
1B Hidden W95 FAT32     85 Linux extended  EF EFI (FAT-12/16/32)
1C Hidden W95 FAT32 (LB 86 NTFS volume set  F0 Linux/PA-RISC boot
1E Hidden W95 FAT16 (LB 87 NTFS volume set  F1 SpeedStor
24 NEC DOS              8E Linux LUM      F4 SpeedStor
39 Plan 9              93 Amoeba        F2 DOS secondary
3C PartitionMagic recov 94 Amoeba BBT     FD Linux raid autodetec
40 Unix 80286          9F BSD/OS        FE LANstep
41 PPC PReP Boot       A0 IBM Thinkpad hiberna FF BBT
42 SFS                 A5 FreeBSD
4D QNX4.x              A6 OpenBSD

Enter filesystem type: 83_

```

Ripetiamo i vari passaggi per ogni ulteriore partizione, naturalmente senza indicare queste nuove partizioni come avviabili (fra l'altro, Windows parte solo se collocato nella prima partizione del disco fisso, che dev'essere necessariamente avviabile). Quindi diamo l'ordine di scrivere andando coi tasti freccette alla voce «write»:

```

cfdisk 2.12
Disk Drive: /dev/hda
Size: 8589934592 bytes, 8589 MB
Heads: 255 Sectors per Track: 63 Cylinders: 1044

Name      Flags      Part Type  FS Type    [Label]    Size (MB)
-----
hda1     Primary   Linux      Free Space
          Pri/Log
          Free Space

[Bootable] [ Delete ] [ Help ] [Maximize] [ Print ]
[ Quit ] [ Type ] [ Units ] [ Write ]

Write partition table to disk (this might destroy data)_

```

e confermiamo con «yes»:

```

cfdisk 2.12
Disk Drive: /dev/hda
Size: 8589934592 bytes, 8589 MB
Heads: 255 Sectors per Track: 63 Cylinders: 1044

Name      Flags      Part Type  FS Type    [Label]    Size (MB)
-----
hda1     Primary   Linux      Free Space
          Pri/Log
          Free Space

Are you sure you want write the partition table to disk? (yes or no): yes_
Warning!! This may destroy data on your disk!

```

Attendiamo quindi il compimento dell'operazione:

```

cfdisk 2.12

Disk Drive: /dev/hda
Size: 8589934592 bytes, 8589 MB
Heads: 255 Sectors per Track: 63 Cylinders: 1044

-----
Name      Flags      Part Type  FS Type    [Label]    Size (MB)
-----
hda1     Primary   Linux     Linux      [Label]    5000.98
          Pri/Log   Free Space
          Pri/Log   Free Space    3586.23

Are you sure you want write the partition table to disk? (yes or no): yes

Writing partition table to disk..._

```

Continuiamo con la stessa procedura per tutte le partizioni nuove che intendiamo creare (ricordando che la scrittura può essere effettuata un'unica volta, al termine di tutte le modifiche):

```

cfdisk 2.12

Disk Drive: /dev/hda
Size: 8589934592 bytes, 8589 MB
Heads: 255 Sectors per Track: 63 Cylinders: 1044

-----
Name      Flags      Part Type  FS Type    [Label]    Size (MB)
-----
hda1     Primary   Linux     Linux      [Label]    5000.98
          Pri/Log   Free Space
          Pri/Log   Free Space    3586.23

[ Help ] [ New ] [ Print ] [ Quit ] [ Units ]
[ Write ]

Create new partition from free space_

```

Una volta che il programma abbia terminato la scrittura, usciamo e riavviamo il sistema. Ciò non è strettamente necessario, ma non sempre l'operazione di `ioctl` che causa la rilettura della tabella delle partizioni va buon fine. A questo punto il comando: `fdisk -l` dovrebbe farci vedere tutte le nuove partizioni. Come alternativa, possiamo lasciare lo spazio libero non allocato e impostare le partizioni direttamente, durante l'installazione, sempre con `cfdisk`.

In caso di errore, possiamo sempre uscire dal programma senza scrivere, scegliendo la voce «Quit»:

```

cfdisk 2.12

Disk Drive: /dev/hda
Size: 8589934592 bytes, 8589 MB
Heads: 255 Sectors per Track: 63 Cylinders: 1044

-----
Name      Flags      Part Type  FS Type    [Label]    Size (MB)
-----
hda1     Primary   Linux     Linux      [Label]    5000.98
hda2     Primary   Linux swap
          Pri/Log   Free Space
          Pri/Log   Free Space    3076.26

[Bootable] [ Delete ] [ Help ] [Maximize] [ Print ]
[ Quit ] [ Type ] [ Units ] [ Write ]

Quit program without writing partition table_

```

2.2.3 Formattare le partizioni

Ora dobbiamo formattare le partizioni create per dotarle di un file system. Ci sono comandi diversi a seconda del file system prescelto (vedi **Scheda n. 3 – I file system di GNU/Linux**):

```
mkfs.ext2 /dev/nome_dispositivo = crea una partizione EXT2.
```

```
mkfs.ext3 /dev/nome_dispositivo
```

oppure

```
mke2fs -j /dev/nome_dispositivo = creano una partizione EXT3.
```

```
mkreiserfs /dev/nome_dispositivo = crea una partizione Reiserfs,
```

dove *nome_dispositivo* corrisponde alla partizione prescelta del disco fisso (dunque hda1 ecc.).

Per passare da EXT2 a EXT3 (operazione che, in fondo, si riduce ad aggiungere un file di journal al file system), basta dare il comando: `tune2fs -j /dev/hdn`, senza dimenticare di modificare `/etc/fstab` in relazione a questo cambiamento (anche se, passando da EXT2 a EXT3 la partizione verrà montata comunque, sebbene senza journaling, e saremo avvertiti di questo dai messaggi di boot, cioè nella fase di caricamento del sistema).

Scheda n. 3

I file system di GNU/Linux

GNU/Linux supporta una grandissima quantità di file system, nativi o utilizzati da altri sistemi operativi, fra cui alcuni assai poco diffusi e particolari. I primi sono supportati direttamente nel kernel (EXT2, EXT3, NFS, iso9660), gli altri grazie a moduli. Ne offriamo un elenco per categorie, distinguendo i file system locali da quelli di rete, e quindi soffermandoci su quelli nativi più usati.

Il journaling

Preliminarmente, va chiarito che il journaling di cui sono dotati alcuni file system serve a conservare le tracce dell'attività del disco fisso: in caso di crash, si tratta di verificare quanto il file system è lontano dalla consistenza e, se possibile, di correggerne gli errori; una ulteriore possibilità, non sempre fortunata, è cercare di recuperare i dati perduti nel crash. L'operazione di scansione e di recupero dei dati è detta *replay*. Il journaling può essere effettuato sui soli metadata oppure sia sui metadata sia sui dati. Nel primo caso è relativamente semplice ripristinare la consistenza del file system, ma nulla garantisce che non vadano persi dei dati. Nel secondo caso, dovendo i dati essere scritti due volte, una nel log di sistema e una nel file system stesso, le prestazioni del sistema si rallentano in misura considerevole. L'unico file system che riesce a compiere questa doppia operazione senza influenzare significativamente le prestazioni del sistema sembra essere Reiserfs4.

Quale file system?

Per vedere quali file system sono supportati dal nostro sistema (il che dipende dalle opzioni con cui è stato compilato il kernel) si usa il comando: `cat /proc/filesystems`.

Non esiste una risposta univoca alla domanda: quale file system adottare per il nostro computer? In linea di massima, attualmente la scelta è fra EXT2, EXT3 e Reiserfs. Dobbiamo considerare elementi quali velocità, stabilità e supporto in funzione delle nostre esigenze, dal momento che un computer domestico deve offrire prestazioni del tutto diverse da un server di posta o da un database. Inoltre va tenuto presente che (a eccezione della migrazione da EXT2 a EXT3 e viceversa), cambiare il file system significa dover formattare e reinstallare i contenuti del nostro disco fisso. In certi ambiti, poi, non vi sono prestazioni significativamente diverse fra i vari file system, mentre EXT2 ed EXT3 risultano quelli più supportati, se ci riferiamo alle utilità di backup, ripristino e gestione. Va infine ricordato che non tutti i bootloader funzionano sempre e comunque (al momento attuale Grub riesce a supportare Reiserfs4 solo se patchato e Lilo non lo supporta affatto). EXT2, proprio perché manca del journaling, ha talora prestazioni migliori di EXT3, ma richiede tempi assai più lunghi per i controlli in caso di crash. Alcuni

ritengono che la scelta migliore per i desktop sia Reiserfs (che Slackware ci propone come default e che è adottato personalmente da P. J. Volkerding), molti altri adottano EXT2 o EXT3. Questi ultimi gestiscono con maggiore efficienza file di notevoli dimensioni, laddove Reiserfs è più veloce nel trattamento dei file di piccole dimensioni (che però sono in genere assai più piccoli di quelli che può creare un utente domestico per un normale uso del computer).

File system locali

bfs file system di boot per SCO Unixware. Non supporta molte funzioni normali, appunto perché creato esclusivamente per bootare il kernel da UnixWare.

cramfs file system compresso residente in ROM.

ext2 file system tipico di GNU/Linux. Offre ancor oggi prestazioni buone e un'ampia affidabilità. È larghissimamente supportato dai sistemi Unix e può valersi di un'ampia serie di utilità e di driver. Non presenta funzioni di journaling, per cui, a seguito di un crash, la sua consistenza dev'essere verificata con `fsck` e può causare la perdita irreparabile di dati. Può essere convertito senza problemi in EXT3.

ext3 è la versione journalized di EXT2. È dotato di un journaling completo (cioè non dei soli metadata), per cui è in genere possibile ripristinare file cancellati o perduti a seguito di un crash. Può essere riconvertito senza problemi in EXT2. Offre tre metodi di journaling: **writeback** (scrive prima i metadata e quindi i dati; la persistenza di dati vecchi può far sì che dopo un ripristino essi compaiano nei file in luogo di quelli più recenti); **ordered** (si tratta dell'impostazione di default: scrive prima i dati e quindi i metadata); **journal** (effettua una doppia scrittura di tutti i dati). La modalità di default può essere modificata inserendo in `/etc/fstab` l'opzione `data=opzione`. EXT3 può memorizzare file fino a 2 TB col kernel 2.4 e fino a 4 TB col kernel 2.6, e inoltre supporta nomi di file fino a 255 caratteri (anche se questo limite può essere superato con opportune patch).

fat file system Microsoft di tipo FAT12 e FAT16. Prende il nome dalla File Allocation Table, una tabella che funge da indice del contenuto del disco fisso. È stato progressivamente perfezionato per aggiungergli nuove funzionalità, cosicché sono nati gli altri, successivi file system della Microsoft. Supporta solo nomi brevi dei file (otto caratteri, più i tre dell'estensione).

freevxfs versione libera del file system Veritas VxFS usato da SCO: Unixware, Sun Solaris, HP-UX. È dotato di journaling.

hfs file system gerarchico Macintosh, oggi aggiornato con hfs+. È supportato da GNU/Linux solo in maniera largamente sperimentale.

jfs file system IBM con funzionalità di journaling. Sviluppato originariamente da IBM, dal 2000 ne esiste anche il port per GNU/Linux. Supporta il journal dei soli metadata e non è in grado di gestire le quote di GNU/Linux.

minix file system Minix. Si tratta, storicamente, del primo file system supportato da GNU/Linux e presenta notevoli limitazioni.

ntfs file system NT Microsoft per Windows NT, Windows 2000 e Windows XP. È sorto dal progetto Microsoft di creare un file system del tutto nuovo, più rapido, resistente ai crash di Windows ed esente dalle limitazioni del file system FAT.

reiserFS file system Reiser, creato *ex novo* da Hans Reiser. Reiserfs è un file system relativamente recente, molto veloce, che organizza le informazioni in un database ad albero bilanciato, per cui è in grado

di utilizzare e trovare anche un grandissimo numero di file di piccole dimensioni in tempi molto rapidi, mentre gestisce con maggior lentezza di EXT2 e di EXT3 i file più ampi. Fa il journaling dei soli metadata: questo vuol dire che in caso di crash o di spegnimento accidentale del PC, le modifiche effettuate sui nostri file potrebbero comunque andare perse. Non è compatibile con EXT2 ed EXT3.

reiserfs4 file system Reiser di nuova generazione. Rilasciato molto di recente, è stato dichiarato il più veloce file system per GNU/Linux. È tutto giocato su operazioni atomiche: dunque esse o sono eseguite completamente o non hanno affatto luogo e non c'è nessuna perdita di dati per operazioni eseguite solo parzialmente. Reiserfs4 organizza il database dei file in alberi che non distinguono file piccoli e grandi, e ciò consente sia una maggior velocità nelle prestazioni sia un notevole risparmio di spazio. Questo file system, a causa della sua novità, non è ancora supportato in maniera nativa dal kernel di GNU/Linux (e necessita pertanto di patch).

romfs file system per ROM. Si tratta di un file system di sola lettura, concepito per gestire i RAM disk dei dischi di installazione.

sysv file system System V Coherent adottato da Xenix, System V, Coherent.

ufs file system Unix usato per i sistemi BSD, Sun e Mac OS X.

umsdos più che un filesystem, si tratta di un layer che permette di implementare i permessi classici dei filesystem Unix sulla FAT, rendendo possibile l'installazione di GNU/Linux su quest'ultima.

vfat file system FAT di Microsoft, noto anche come FAT32. Questo file system, a differenza dei precedenti MSDOS, supporta nomi fino a 255 caratteri.

xfs file system creato da SGI per Irix (una distribuzione Unix). Dal 2000 ne esiste anche il port per GNU/Linux. Ha caratteristiche tecniche originali, come la struttura ad albero e il supporto per attributi estesi (oggi posseduto da diversi altri file system), e garantisce una larghezza di banda minima durante l'accesso.

File system di rete

nfs file system NFS (Network File System). Sviluppato da Sun e basato sul meccanismo RPC, consente la condivisione dei file in sistemi Unix paritetici e dunque anche GNU/Linux (sebbene esista un add-on della Microsoft per usare NFS anche su Windows). Essendo stato concepito per reti interne, non possiede funzionalità di protezione integrata.

coda file system di rete della Carnegie Mellon University. Simile a NFS, offre anche il supporto per operazioni interrotte e funzioni di cache di sicurezza.

interMezzo file system simile a Coda, gestito come progetto GPL.

ncfps file system NCFPS di Novell Corporation (detto anche NCP). Serve a montare i volumi Novell Netware.

netatalk file system atto a gestire le reti miste GNU/Linux e Macintosh, sia paritetiche sia client.

smbfs file system basato sul protocollo SMB (Shared Message Block) usato da Microsoft e Samba stesso. Consente di gestire servizi di rete ed è stato sviluppato come implementazione SMB/CIFS. L'implementazione Unix prende il nome di Samba ed è comunemente usata per gestire le reti miste GNU/Linux e Microsoft Windows od OS/2, sia paritetiche sia client.

File system per CD-ROM

iso 9960 è lo standard per i CD-ROM; in particolare, permette la registrazione in multisessione. Per soddisfare particolari esigenze, sono state create per esso nuove estensioni.

rock Ridge consente l'uso di nomi lunghi e link simbolici di tipo UNIX. Appartenendo tuttavia allo standard ISO 9960, può essere usato anche nei sistemi non Unix.

joliet estensione di ISO 9960 sviluppata da Microsoft. Consente l'uso dei caratteri Unicode e dei nomi lunghi (fino a un massimo di 64 caratteri).

udf è lo standard (ISO 13346) per i DVD-ROM (acronimo di Universal Disk Format). Consente la scrittura e la cancellazione di un CD-RW senza le perdite previste dalla modalità multisessione. Supporta grandi dimensioni di file e tecniche di scrittura a pacchetti, per cui il CD può essere utilizzato come una sorta di disco fisso.

Più ampie informazioni su diversi file system possono essere reperite innanzitutto in: `//usr/src/linux/Documentation/filesystems`.

2.2.4 Il file di swap

Il file di swap, che è un'intera partizione, va creato col comando: `mkswap /dev/nome_dispositivo`. Una volta creata, tale partizione dev'essere attivata col comando: `swapon /dev/nome_dispositivo`.

Invece di una partizione, sarebbe possibile creare lo swap come un semplice file all'interno di una preesistente partizione (operazione che torna utile solo in casi particolari, di emergenza, dato inoltre che la lettura di tale file è estremamente più lenta di quella della partizione). In questo caso bisogna creare un file vuoto specificandone la dimensione in MB (*numero_MB*), inizializzarlo come file di swap e infine aggiungerlo al pool di swapping. I tre comandi relativi sono:

```
1) dd if=/dev/zero of=swap_file bs=1024k count=numero_MB
```

```
2) mkswap swap_file
```

```
3) swapon swap_file.
```

2.2.5 Le partizioni DOS

Aggiungiamo, a titolo informativo, che le partizioni DOS si formattano col comando `mkdosfs`. Ad esempio, una semplice partizione DOS si crea col comando: `mkdosfs /dev/nome_dispositivo`, mentre una partizione FAT32 (vfat) si crea col comando: `mkdosfs -F32 /dev/nome_dispositivo`.

2.2.6 Una partizione di scambio

Nel caso si debbano trasferire dati da una partizione Windows NTFS a una GNU/Linux e viceversa, è necessario creare un'ulteriore partizione di scambio in FAT32. GNU/Linux infatti legge e scrive regolarmente su FAT32, mentre è in grado di leggere soltanto il file system NTFS. Col kernel 2.6 è stata introdotta anche la possibilità di scrivere su NTFS, ma con fortissime limitazioni che ne rendono sconsigliabile – e nella maggior parte dei casi inutile – l'utilizzo. Va ricordato che le partizioni FAT32 supportano solo in forma statica i permessi, assegnati una volta per tutte in `/etc/fstab` con `umask`, e non supportano invece i link simbolici. Per questo va sempre evitato di effettuare operazioni come la scom-

pattazione dei file, e ancor meno tentativi di compilazione o installazioni di programmi su partizioni di questo tipo¹.

2.2.7 Esempi di configurazione

Ipotizziamo, a titolo esemplificativo, di scegliere la configurazione minimale (secondo alcuni, sarebbe inopportuno avere più di due partizioni primarie, per cui le altre andrebbero attivate come logiche all'interno di una estesa) e avere un solo disco fisso contenente le seguenti partizioni:

- hda1: partizione primaria formattata NTFS contenente Windows
- hda2: partizione primaria formattata EXT3, per installarci Slackware
- hda3: partizione primaria formattata SWAP, per il file di swap
- hda4: partizione primaria formattata FAT32, per il file di scambio.

Nel caso si scelga anche una partizione distinta per /home, si avrà una partizione estesa contenente due partizioni logiche, al che il nostro disco fisso potrà apparire come segue:

- hda1: partizione primaria formattata NTFS contenente Windows
- hda2: partizione primaria formattata EXT3, per installarci Slackware
- hda3: partizione primaria formattata SWAP, per il file di swap
- hda4: partizione estesa contenente hda5 e hda6
- hda5: partizione logica formattata FAT32, per il file di scambio
- hda6: partizione logica formattata EXT3, per installarci la home dell'utente.

Per GNU/Linux, hda è il disco master sul primo canale ATA, cioè IDE1, e hdb il disco slave sullo stesso canale; hdc è il disco master sul secondo canale ATA, cioè IDE2, e hdd il disco slave sullo stesso canale. Le partizioni sono i numeri che seguono la sigla: hda1 è la prima partizione del disco fisso master, hda2 la seconda; hdb1 è la prima partizione sul disco fisso slave, e così via. Secondo alcune scuole di pensiero, sarebbe opportuno non collegare due dischi fissi allo stesso canale, per evitare interferenze che possono produrre errori nella trasmissione dei dati (ciò è ancora più vero se ci si riferisce a unità come lettori CD e masterizzatori, che se collegati allo stesso canale possono determinare il fallimento delle copie «al volo»); tuttavia, in genere, i dischi fissi sono collegati al primo canale e dispositivi come lettore CD, masterizzatore e unità DVD al secondo.

I dischi SATA vengono invece riconosciuti come dispositivi SCSI, per cui prendono il nome di sda, sdb ecc. Le partizioni saranno di conseguenza sda1, sda2, ..., sdb1 ecc. Unità IDE come lettori CD, masterizzatori e DVD saranno riconosciuti in questo caso come hda ecc.; se si attiva l'emulazione SCSI, tali unità prenderanno il nome di sr0 ecc., oppure scd0 ecc.; di norma, per rendersene conto è sufficiente leggere le righe pertinenti di con `dmesg | more`. Si può tentare di identificare subito il dispositivo con `dmesg | grep sr0` e simili.

Ricordiamo incidentalmente che st0 ecc. sono le unità di memorizzazione a nastro e sg0 ecc. tutti i dispositivi che non ricadono nelle fattispecie sopra descritte.

¹Non analizziamo qui né la procedura per la ricompilazione del kernel 2.4, né quella per la compilazione del nuovo kernel 2.6. A tale proposito, due guide estremamente chiare e precise si possono leggere ai seguenti indirizzi: http://www.slacky.it/tutorial/kernel/compilazione_kernel.html e <http://www.gentoo.it/tips/kernel-2.6.0.html>.

Scheda n. 4**La partizione /home**

GNU/Linux è un sistema autenticamente multiutente: ciò significa che prevede l'accesso contemporaneo allo stesso computer da parte di utenti diversi, che non devono poter accedere ai dati altrui. L'utente root (chiamato anche «superuser») è responsabile della gestione del sistema, può modificare qualsiasi cosa nel sistema e può accedere alle partizioni dei vari utenti. Ciascun utente non root (detto *user*) non può modificare il sistema, e possiede uno «spazio», appunto la sua home, dove lavora, può memorizzare i suoi dati, personalizzare i programmi che usa senza che queste modifiche si rispecchino nelle home degli altri utenti. Il nome dell'utente dovrebbe essere inferiore agli otto caratteri, per compatibilità con le utility di login.

Se anche esiste un solo utente che usa il computer, il dualismo di root e utente è vantaggioso perché, se si riserva a root di modificare il sistema e si lavora di norma come utente, è più difficile danneggiare il sistema stesso. In secondo luogo, nel caso si debba eliminare momentaneamente il sistema operativo in vista di una reinstallazione, i dati memorizzati in home rimarranno inalterati e così pure le impostazioni. Infine eventuali virus (eventualità allo stato attuale abbastanza remota in GNU/Linux) non possono danneggiare il sistema, perché non hanno accesso alle sue parti vitali. Per creare e montare correttamente la partizione di home in un'altra partizione o in un altro disco fisso, per essa andrà indicato /home come punto di montaggio e non invece /mnt/home.

Capitolo 3

Installare Slackware

3.1 Avviare l'installazione

Acceso il computer e inserito il CD 1 (che contiene tutti i file vitali per il funzionamento del sistema operativo), ha luogo il boot da CD (ovvio, ma da non dimenticare: il BIOS del computer dev'essere settato in modo da consentire il boot da CD). Il programma di installazione chiede innanzitutto di scegliere un kernel della serie 2.4.33.3: ce ne sono numerosi (vedi [Scheda n. 5 – I kernel di Slackware 11.0](#)), per rispondere a svariate esigenze di configurazione hardware (l'elenco completo dell'hardware supportato si legge nel file README.TXT presente nella directory /bootdisk del CD 2). In generale, i kernel con l'estensione .i servono per sistemi con controller IDE, mentre quelli con l'estensione .s supportano in aggiunta controller SCSI. Il supporto nativo a SATA è *build-in* solo in alcuni di questi kernel, mentre gli altri richiedono un initrd. La motivazione è essenzialmente pratica: tali kernel devono poter stare in un floppy di 1.44 MB, il che è possibile solo per kernel largamente modulari. L'elenco che precisa quali kernel sono compilati con tale supporto *build-in* e quelli che invece lo implementano in forma solo modulare si legge in /kernels/SATA-SUPPORT.txt.

Scheda n. 5

I kernel di Slackware 11.0

Slackware 11.0 ci mette a disposizione un'ampia serie di kernel precompilati della serie 2.4.33.3. I seguenti supportano sistemi IDE, e alcuni, in aggiunta, alcuni controller SATA:

ataraid.i kernel dotato di supporto per controller IDE RAID.

bare.i è il kernel di base, valido per la maggior parte dei PC con controller di tipo IDE e supporta CD-ROM e DVD IDE/ATAPI.

bareacpi.i kernel come **bare.i**, con l'aggiunta del supporto per ACPI (Advanced Configuration and Power Interface). È così in grado di svolgere funzioni come l'APM (Advanced Power Management). Solitamente, si usa sui laptop.

lowmem.i kernel utile per sistemi IDE con scarsa RAM (meno di 8 MB), che inoltre supporta i processori i386.

old_cd.i kernel analogo al **bare.i**, con l'aggiunta di un supporto per una serie di vecchi lettori CD-ROM non standard.

pportide.i kernel analogo al **bare.i**, con un più ampio supporto per dispositivi IDE su porta parallela.

sata.i kernel analogo al **bare.i**, con supporto per controller SATA di Promise, Silicon Image, SiS, ServerWorks / Apple K2, VIA e Vitesse. È presente a partire da Slackware 10.1.

I seguenti supportano sistemi dotati di controller SCSI:

adaptec.s kernel che supporta un'ampia serie di controller SCSI Adaptec

ibmmca.s kernel che supporta la MicroChannel Architecture, presente in alcuni computer IBM PS/2 e in alcuni laptop. Include inoltre il supporto per svariati adattatori MCA SCSI, Ethernet e Token Ring

jfs.s kernel come il **bare.i**, ma col supporto per l'IBM Journaled file system e per l'Adaptec AIC7xxx

raid.s kernel col supporto per alcuni controller SCSI e ATA RAID

scsi.s kernel col supporto per svariati controller SCSI non Adaptec

scsi2.s kernel col supporto per altri controller SCSI non Adaptec

scsi3.s kernel col supporto per altri controller SCSI non Adaptec

speakup.s come il **bare.i**, ma col supporto per Speakup (strumento per i non vedenti, in quanto invia gli output della console a diversi sintetizzatori vocali)

xfs.s versione estesa di **bare.i**, col supporto per il file system XFS (SGI)

huge26s kernel 2.6.17.13, non ancora considerato come scelta di default

Qui prendiamo in considerazione solo l'ipotesi di un hardware senza componenti particolari (come dischi o masterizzatori SCSI o dischi fissi e relativi controller SATA, nel quale caso dovrebbe comunque essere sufficiente scegliere all'avvio del processo di installazione il kernel **sata.i** e riconfermare la scelta verso la conclusione). Nel caso di incertezze, il tasto **F2** fornisce un elenco più ampio di kernel, e quindi il tasto **F3** ne offre l'elenco completo. Nel caso che consideriamo, al prompt ci si limita a scegliere la modalità proposta di default, e cioè il kernel **bare.i**. È sufficiente dare **Invio** al prompt:

```
ISOLINUX 2.10 2004-06-18 Copyright (C) 1994-2004 H. Peter Anvin
Welcome to Slackware version 10.0 (Linux kernel 2.4.26)!

If you need to pass extra parameters to the kernel, enter them at the prompt
below after the name of the kernel to boot (scsi.s etc). NOTE: In most cases
the kernel will detect your hardware, and parameters are not needed.

Here are some examples (and more can be found in the BOOTING file):
  hdx=cyls,heads,sects,mpcom,irq (needed in rare cases where probing fails)
or hdx=cdrom (force detection of an IDE/ATAPI CD-ROM drive)
where hdx can be any of hda through hdt.

In a pinch, you can boot your system from here with a command like:

For example, if the Linux system were on /dev/hda1.

boot: bare.i root=/dev/hda1 noinitrd ro

This prompt is just for entering extra parameters. If you don't need to enter
any parameters, hit ENTER to boot the default kernel "bare.i" or press [F2]
for a listing of more kernel choices.

boot: _
```

Slackware dà la possibilità di installare anche un kernel della serie 2.6, e precisamente il 2.6.17.13. Esso non viene considerato più in testing, in quanto ha ormai raggiunto una buona stabilità (sebbene,

a quanto sembra, Volkerding continua a preferire i kernel della serie 2.4 almeno negli ambienti di produzione per la loro ormai diuturna affidabilità), può nondimeno rappresentare una buona scelta per un computer desktop, data la sua minor latenza e il più ampio supporto hardware. In questo caso, al prompt andrà scritto: `huge26.s`. Questo kernel supporta nativamente anche i dischi SATA e SCSI e presenta il supporto per la «high memory», utile ai computer che hanno più di un GB di RAM. In un secondo momento, al termine dell'installazione e prima di riavviare il sistema, andranno installati anche i moduli ed eventualmente gli header del kernel.

3.2 Scegliere la tastiera

In secondo luogo dobbiamo scegliere la tastiera. Visto che siamo italiani, se non ci va bene quella di default, che è quella inglese statunitense, battiamo **1** per cambiarla:

```
LUM version 1.0.0(17/11/2003)
Initializing Cryptographic API
NET4: Linux TCP/IP 1.0 for NET4.0
IP Protocols: ICMP, UDP, TCP, IGMP
IP: routing cache hash table of 2048 buckets, 16Kbytes
TCP: Hash tables configured (established 16384 bind 32768)
Linux IP multicast router 0.06 plus PIM-SM
NET4: Unix domain sockets 1.0/SMP for Linux NET4.0.
RAMDISK: Compressed image found at block 0
Freeing initrd memory: 2607k freed
UFS: Mounted root (ext2 filesystem).
Freeing unused kernel memory: 128k freed
init started: BusyBox v0.60.5 (2003.02.16-05:06+0000) multi-call binary
proc on /proc type proc (rw)
Probing for USB devices.
(to skip, give a 'nouseb' kernel option at boot)

<OPTION TO LOAD SUPPORT FOR NON-US KEYBOARD>

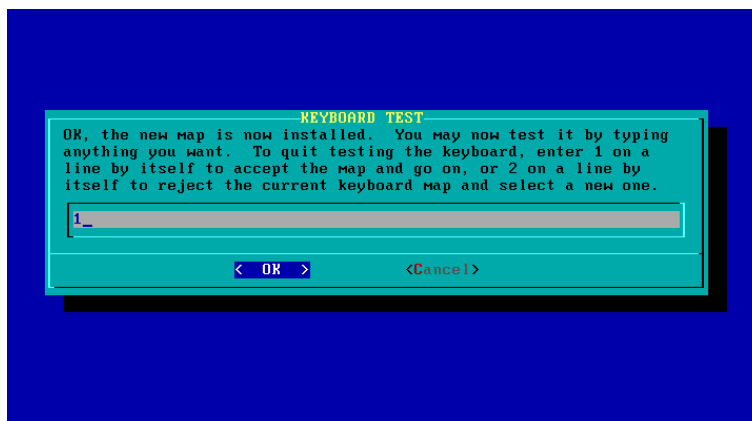
If you are not using a US keyboard, you may now load a different
keyboard map. To select a different keyboard map, please enter 1
now. To continue using the US map, just hit enter.

Enter 1 to select a keyboard map: 1_
```

Nel menu che ci compare selezioniamo **qwerty/it.map**:

```
KEYBOARD MAP SELECTION
You may select one of the following keyboard maps.
If you do not select a keyboard map, 'us.map' (the
US keyboard map) is the default. Use the UP/DOWN
arrow keys and PageUp/PageDown to scroll through
the whole list of choices.
↑(+)-
qwerty/gr.map
qwerty/hu101.map
qwerty/il-heb.map
qwerty/il-phonetic.map
qwerty/il.map
qwerty/is-latin1-us.map
qwerty/is-latin1.map
qwerty/it-ibm.map
qwerty/it.map
qwerty/it2.map
qwerty/jp106.map
↓(-)
< OK > <Cancel>
```

e diamo **Invio**. Poi confermiamo la scelta, sempre col tasto **1** seguito da **Invio**:



3.3 Lanciare il setup

Ora possiamo fare il login (l'ingresso nel sistema) come root (che è l'amministratore del sistema), per avviare il setup vero e proprio al prompt (che appare: root@slackware: /#) e digitando la stringa root:

```
Welcome to the Slackware Linux installation disk! (version 10.0)
##### IMPORTANT! READ THE INFORMATION BELOW CAREFULLY. #####
- You will need one or more partitions of type 'Linux' prepared. It is also
  recommended that you create a swap partition (type 'Linux swap') prior
  to installation. For more information, run 'setup' and read the help file.
- If you're having problems that you think might be related to low memory (this
  is possible on machines with 16 or less megabytes of system memory), you can
  try activating a swap partition before you run setup. After making a swap
  partition (type 02) with cfdisk or fdisk, activate it like this:
  mkswap /dev/<partition> ; swapon /dev/<partition>
- Once you have prepared the disk partitions for Linux, type 'setup' to begin
  the installation process.
- If you do not have a color monitor, type: TERM=vt100
  before you start 'setup'.
You may now login as 'root'.
slackware login: root_
```

In questa fase non c'è alcuna password da immettere. Visto che abbiamo già preparato il disco fisso, trascuriamo la proposta di formattarlo usando cfdisk o fdisk. Avviamo l'installazione digitando la stringa setup:


```

- If you do not have a color monitor, type: TERM=vt100
  before you start 'setup'.

You may now login as 'root'.

slackware login: root

Linux 2.4.26.

If you're upgrading an existing Slackware system, you might want to
remove old packages before you run 'setup' to install the new ones. If
you don't, your system will still work but there might be some old files
left laying around on your drive.

Just mount your Linux partitions under /mnt and type 'pkgtool'. If you
don't know how to mount your partitions, type 'pkgtool' and it will tell
you how it's done.

To partition your hard drive(s), use 'cfdisk' or 'fdisk'.
To activate PCMCIA/Cardbus devices needed for installation, type 'pcmcia'.
To activate network devices needed for installation, type 'network'.
To start the main installation, type 'setup'.

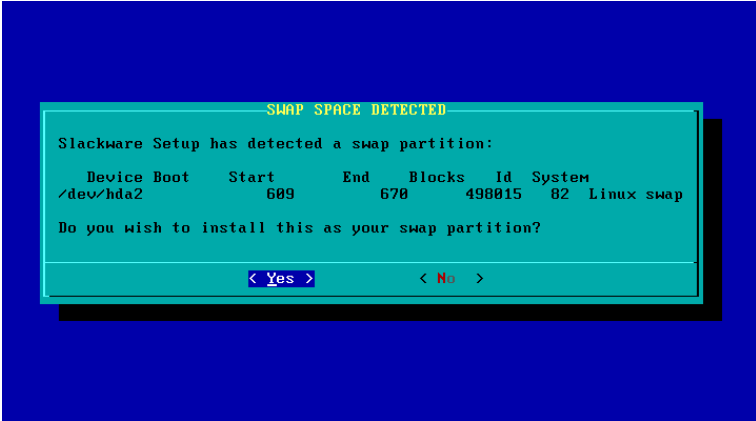
root@slackware:/# setup_

```

Il setup proporrà le seguenti opzioni: **KEYMAP**, **ADDSWAP**, **TARGET**, **SOURCE**, **SELECT**, **INSTALL**, **CONFIGURE**. L'opzione **KEYMAP** non ci interessa, visto che abbiamo già impostato la tastiera.

3.4 Opzione ADDSWAP

Dobbiamo impostare la partizione di swap. Poiché il (primo) disco fisso IDE è identificato come **hda** (sappiamo già che un eventuale secondo disco fisso risulterebbe **hdb**), scegliamo la partizione di swap (nel nostro esempio **/dev/hda2**):



The screenshot shows a terminal window with a blue background. A cyan box contains the following text:

```

SWAP SPACE DETECTED

Slackware Setup has detected a swap partition:

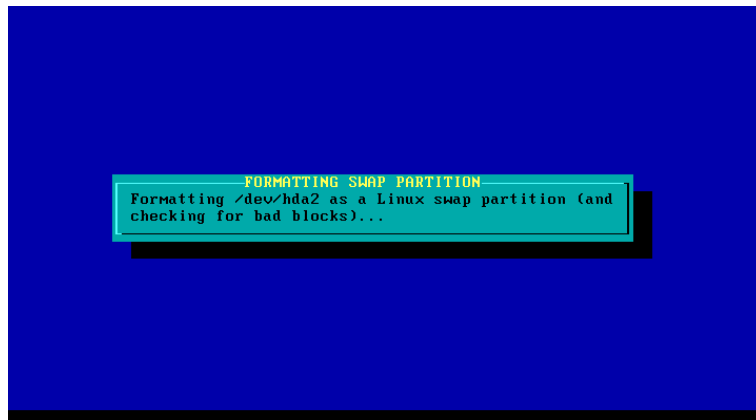
  Device Boot   Start     End  Blocks  Id System
 /dev/hda2     609      670   490015  82 Linux swap

Do you wish to install this as your swap partition?

  < Yes >           < No  >

```

Se l'abbiamo già formattata come tale, Slackware la riconosce automaticamente e ci propone – chiedendoci «Check your swap partitions for bad blocks?» – di riformattarla con o senza il controllo dell'integrità (il quale, di default, non viene compiuto) e quindi la attiva automaticamente col comando **swapon**:

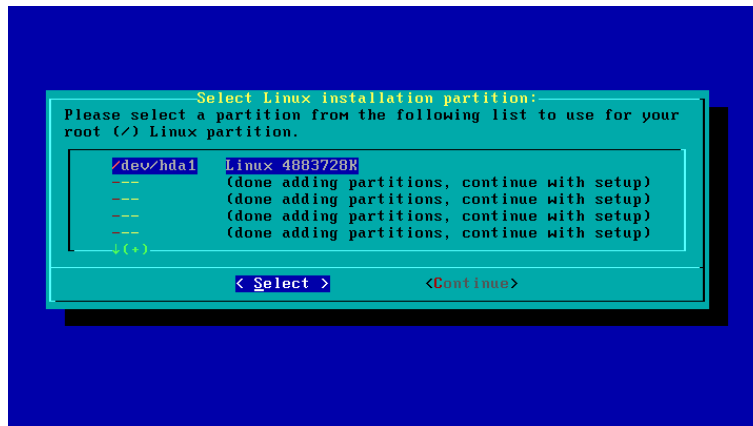


Diamo l'OK quando il programma ci avverte che il file di swap è stato correttamente configurato e aggiunto a `/etc/fstab`:

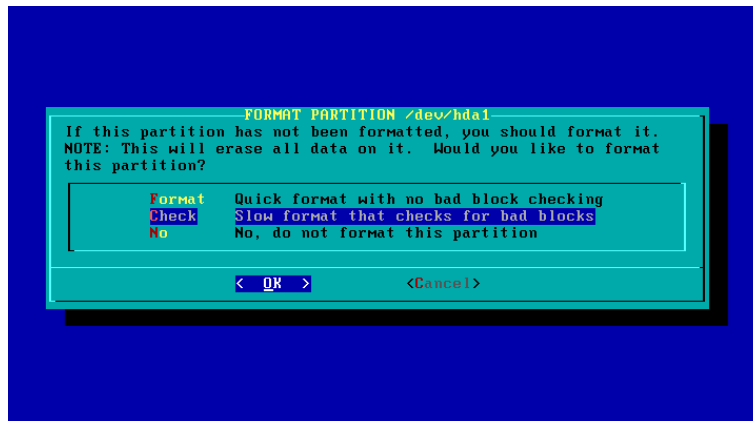


3.5 Opzione TARGET

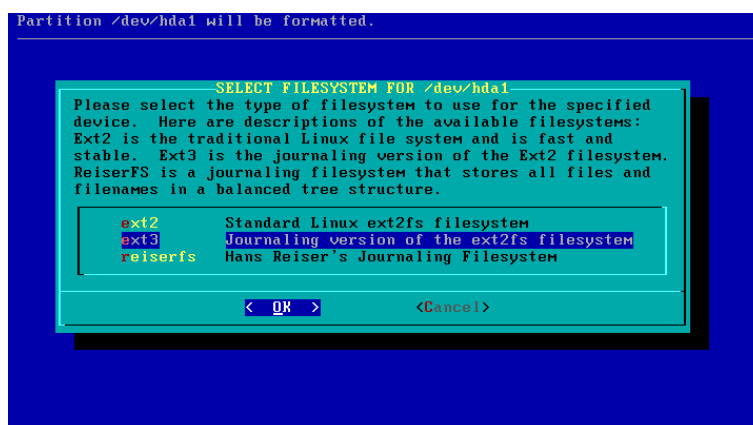
Scegliamo quindi la partizione (o le partizioni) in cui installare il sistema operativo (nel nostro esempio `/dev/hda1`), e battiamo **Invio**. Si tratta della partizione di root, indicata da una semplice barra o *slash* (`/`).



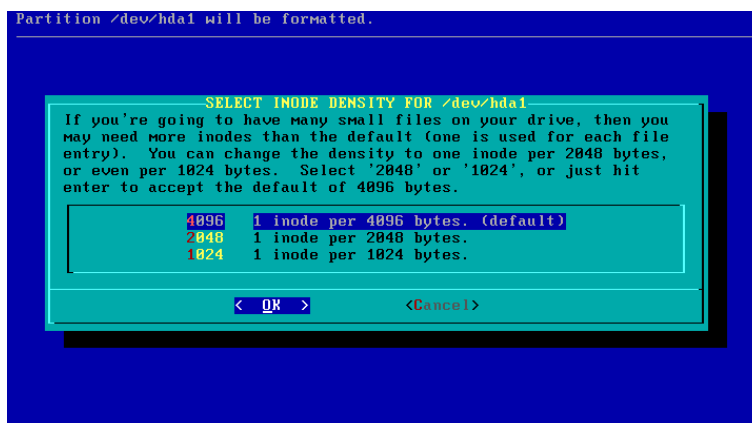
Anche in questo caso, se l'abbiamo già formattata essa verrà riconosciuta subito da uno scanning automatico, ma possiamo effettuare un'ulteriore scelta e cioè mantenere il file system EXT3 oppure riformattarla EXT2 o Reiserfs:



Possiamo ora (ri)formattare la partizione, con o senza controllo, oppure rinunciare alla formattazione:

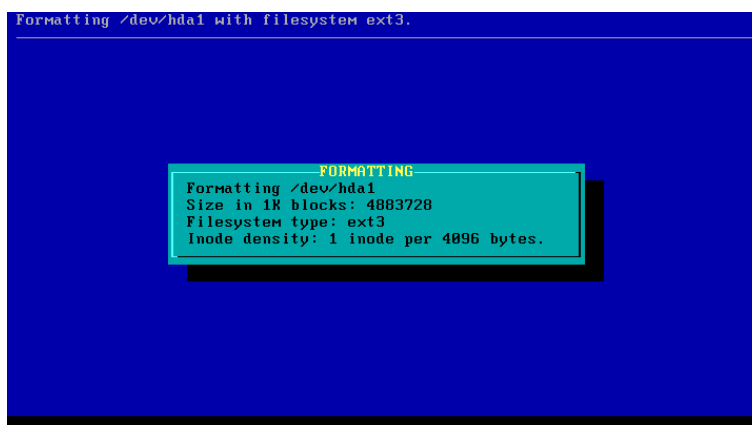


Nel caso si scelga di riformattare nei formati EXT2 o EXT3, il programma chiederà di indicare la densità degli inode (cioè gli «index node» dove vengono registrati i nomi dei file e gli indirizzi dei blocchi in cui sono situati i file):



Ogni file richiede almeno un inode per essere indicizzato, quindi è necessario creare un equilibrio fra le dimensioni del disco fisso e il numero degli inode in base al quale sarà stabilita la quantità di file che è possibile creare (mentre il resto dello spazio è dedicato a blocchi di dati). Di default viene creato un inode ogni 4096 byte, proposta adatta alla maggior parte delle situazioni e che va modificata in 2048 o addirittura in 1024 solo se si prevede di creare nel disco fisso un elevatissimo numero di file molto piccoli.

Slackware propone Reiserfs come default. Anche in questo caso il programma di installazione può effettuare una formattazione veloce oppure con verifica (che è naturalmente più lenta):



Se all'avvio abbiamo scelto il kernel 2.6.17.13 (cioè huge26.s), allora la scelta del file system si fa più ampia: è infatti incluso il supporto non solo per EXT2, EXT3 e Reiserfs, ma anche per jfs e xfs.

Ovvio che, se stiamo utilizzando il programma di installazione solo per apportare delle modifiche a una partizione preesistente e contenente dei dati, sceglieremo di non formattare e saremo avvertiti in questo senso dal messaggio «partition will not be reformatted».

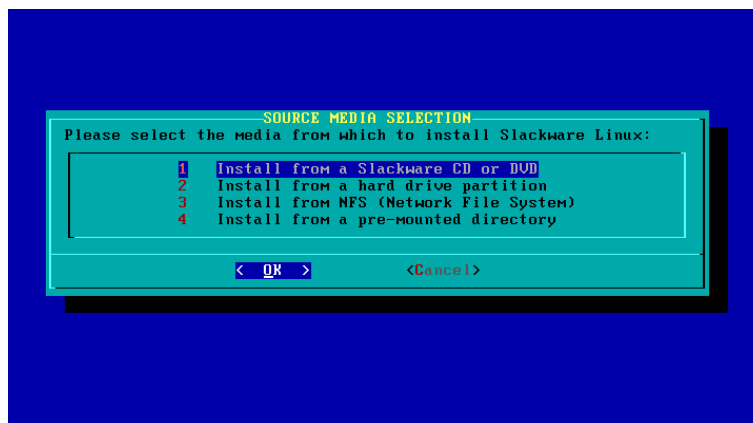
Una volta che la partizione selezionata è stata vista dal programma come «in use», confermiamo che abbiamo terminato di aggiungere partizioni e facciamo scrivere le informazioni in /etc/fstab:



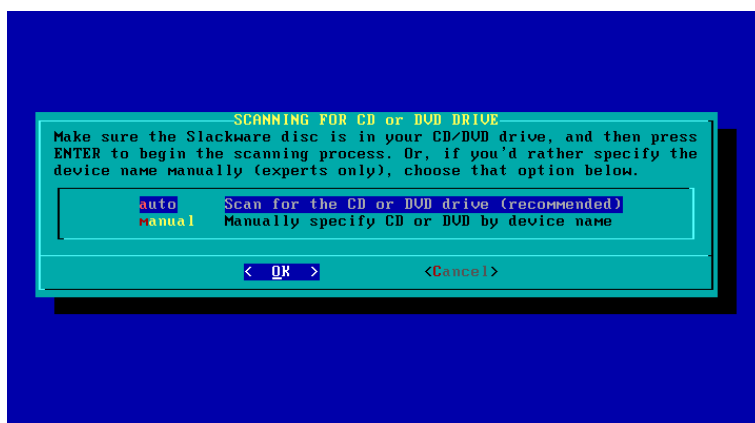
Se invece intendiamo aggiungere altre partizioni, selezioniamo via via la successiva e compiamo le stesse operazioni. Se abbiamo anche partizioni in NTFS o in FAT32, esse vengono rilevate dal programma di installazione, che ci chiede se vogliamo aggiungerle (Slackware 9.1 riconosceva invece solo le partizioni FAT32). In questo caso dobbiamo indicare un punto di montaggio per potervi accedere in seguito (un *pick mount point*): lo indicheremo con un nome a piacere, precisando che, secondo la migliore tradizione Unix, la partizione va montata in /mnt (scriviamo cioè una cosa del tipo: /mnt/nome_della_partizione, come /mnt/Windows).

3.6 Opzione SOURCE

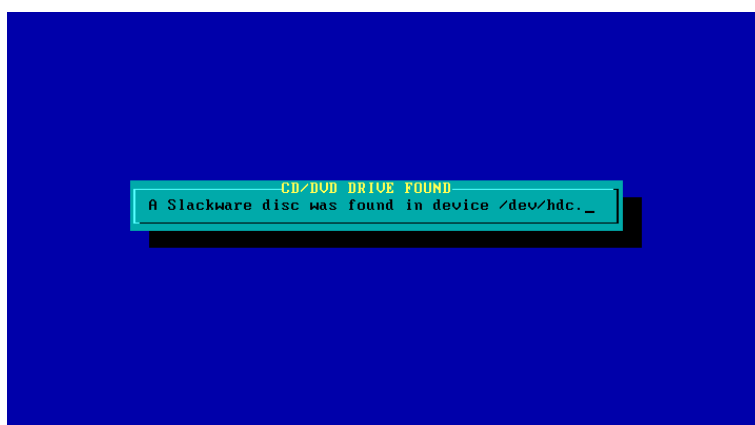
Scegliamo ora la sorgente fisica da cui installare Slackware. Il programma prevede una serie di opzioni: un CD o un DVD, una partizione di disco fisso, NFS (cioè un altro computer in rete) o una directory (da montare manualmente):



Qui consideriamo soltanto il caso di un'installazione da CD o DVD, opzione che pertanto sceglieremo. Il programma di installazione ci chiede di attivare una ricerca automatica del dispositivo o di indicarlo manualmente. Scegliamo la prima soluzione. Quindi si avvia la ricerca della presenza di un CD o DVD di installazione:



Il programma avverte quando il dispositivo viene individuato. Di norma, esso starà nel lettore CD o DVD, identificato come `/dev/hdc` o `/dev/hdd`, anche se ciò dipende dalle porte in cui sono installate queste unità; verosimilmente, in un computer con dischi SATA il lettore potrebbe essere, ad esempio, `/dev/hda`:



3.7 Opzione SELECT

Ora dobbiamo scegliere i programmi da installare nel sistema (per i programmi disponibili in Slackware 11.0, vedi Scheda n. 6 – I pacchetti di Slackware 11.0). Slackware continua a utilizzare una distribuzione dei programmi in pacchetti indicati da lettere dell'alfabeto o da brevi sigle: la cosa ha origine dal fatto che le sue prime versioni risalgono ad anni in cui non erano disponibili i CD-ROM e tanto meno i DVD, per cui i programmi stessi erano distribuiti in una molteplicità di floppy.

Scheda n. 6

I pacchetti di Slackware 11.0

Vediamo in breve che cosa contengono i vari pacchetti presenti in Slackware 11.0 (una descrizione più dettagliata, pacchetto per pacchetto, si legge nel file `PACKAGES.TXT` presente nel CD 1). Nel loro insieme, essi ammontano – decompressi – a circa 3482,4 MB.

Pacchetti della directory /isolinux del CD 1

La directory /isolinux contiene le utilità e tutte le istruzioni atte a creare un set di CD o un DVD di avvio partendo dalla radice dell'albero delle directory di Slackware.

Pacchetti della directory /slackware del CD 1

Serie di pacchetti a (compressi 44,7 MB, decompressi 96,1 MB) è il pacchetto di base (Base Linux System) e contiene tutti i file necessari per una configurazione minimale (e naturalmente priva di strumenti grafici) di Slackware. In teoria sarebbe sufficiente installare questo pacchetto per avere un sistema operativo pienamente funzionante.

Serie di pacchetti ap (compressi 67,1 MB, decompressi 174 MB) contiene applicazioni e programmi che non girano sotto X, cioè senza interfaccia grafica. Sono presenti, fra l'altro, utilità per la masterizzazione, per la stampa, database (MySQL), applicazioni multimediali, le pagine di man, l'editor Vim.

Serie di pacchetti d (compressi 97,7 MB, decompressi 309,3 MB) contiene programmi di sviluppo, come compilatori per i sorgenti (autoconfs, automake), cvs, interpreti di vari linguaggi come gcc (per C e C++), Perl, Python, Ruby ecc. Essenziale agli sviluppatori, serve comunque per gestire Slackware (senza di esso, fra l'altro, non saremmo in grado di compilare alcun pacchetto dai sorgenti).

Serie di pacchetti e (compressi 21,4 MB, decompressi 47,7 MB) contiene l'editor Emacs, che presenta molte altre potenzialità.

Serie di pacchetti f (compressi 8,7 MB, decompressi 30,7 MB) contiene il materiale informativo, e cioè le FAQ (Frequently Asked Questions) e gli How-To.

Serie di pacchetti k (compressi 37,6 MB, decompressi 165,1 MB) contiene i sorgenti del kernel 2.4.33.3.

Serie di pacchetti l (compressi 113,9 MB, decompressi 459,7, MB) contiene le librerie di sistema. Queste servono ovviamente agli sviluppatori, ma sono essenziali per il funzionamento della maggior parte dei programmi. Tale serie presenta, fra l'altro, alcune librerie per gestire i vari formati sonori e delle immagini, le librerie per i programmi in GTK (glib) e per quelli in C (glibc). Pur non integrando Gnome, Slackware contiene le librerie che servono per farne funzionare numerose applicazioni.

Serie di pacchetti n (compressi 62,2 MB, decompressi 156 MB) contiene gli strumenti di networking, per la navigazione in rete. Dunque PPP (per la connessione a Internet), alcuni browser testuali (Lynx), svariati server per il web (Apache), per i servizi DNS (Bind), server e client per la posta elettronica (Fetchmail, Procmil, Mutt, Pine) e per le news (Nn), protocolli di rete (TCP/IP), Samba, svariate utilità per gestire il NFS, per i servizi DHCP e iptables, nonché l'interprete per php, un download manager (Wget) e altro ancora.

Serie di pacchetti x, (compressi 64,5 MB, decompressi 128,3 MB) contiene i file essenziali al server X, cioè il server grafico.

Serie di pacchetti xap (compressi 127,1 MB, decompressi 328,3 MB), prima parte contiene le applicazioni che girano sotto il server grafico. Pertanto include svariati Window Manager (Blackbox, Fluxbox, Fvwm, Fvwm95, Windowmaker), browser (Firefox), client di posta elettronica (Thunderbird), client per chattare (Xchat), server news (Pan), visualizzatori di immagini (Imagemagic, Xv), visualizzatori di file .pdf (Xpdf), programmi di grafica (Sane e Xsane per usare lo scanner), videoscrittura (Xvim), strumenti multimediali (Xmms, Gxine), xscreensaver (raccolta di screensaver) e altro ancora.

Serie di pacchetti y (compressi 2,2 MB, decompressi 5,3 MB) contiene alcuni giochi testuali.

Pacchetti della directory /kernels del CD 1

La directory /kernels contiene i kernel della serie 2.4.33.3 che sono stati visti sopra.

Pacchetti della directory /bootdisk del CD 2

La directory /bootdisk contiene tutti i pacchetti e le utilità atti a creare un floppy di avvio utile nel caso che sia impossibile avviare normalmente il sistema.

Pacchetti della directory /extra del CD 2

La directory /extra (prima parte) contiene una serie di programmi aggiuntivi o alternativi a quelli già presenti nella distribuzione. Fra quelli che sono estremamente utili troviamo:

Bittorrent strumento che utilizza la capacità di upload dei computer che stanno scaricando dei file per redistribuirli ad altri computer in download

Checkinstall utilità per configurare, compilare e creare automaticamente file .tgz (come anche .deb e .rpm) partendo dai sorgenti

Grub il bootloader alternativo a Lilo

J2sdk il pacchetto Java dotato degli strumenti di sviluppo

K3b sontuoso frontend per la masterizzazione di CD e DVD

Parted programma per la gestione delle partizioni

Pacchetti della directory /rootdisk del CD 2

La directory /rootdisk contiene si debba installare Slackware in un sistema che non consente il boot da CD o DVD.

Pacchetti della directory /slackware del CD 2

Serie di pacchetti KDE (compressi 277,4 MB, decompressi 614,9 MB) tutti i pacchetti che compongono KDE e cioè:

amarok lettore multimediale

kdeaccessibility utilità per facilitare la visione dello schermo e l'uso del mouse

kdeaddons plugin e script addizionali per alcune applicazioni di KDE

kdeadmin utilità per la gestione del sistema

kdeartwork temi, screensaver, suoni, sfondi e finestre addizionali

kdebase i file base di KDE. Include kdwin (lo Windows Manager di KDE), konqueror, konsole, kaudio, i file della guida e parecchie altre componenti

kdebindings librerie per i vari linguaggi

kdeedu programmi didattici di KDE

kdegames i giochi di KDE

kdegraphics i programmi di grafica di KDE. Include visualizzatori per vari formati e altri strumenti

kdelibs le librerie necessarie a KDE. Include anche arts, il demone sonoro di KDE

kdemultimedia programmi multimediali di KDE (lettori cd, mp3, mixer, ecc.)

kdenetwork strumenti per la navigazione in rete.

kdepim il KDE Personal Information Manager, e cioè una raccolta di utilità per la gestione delle informazioni personali come la posta elettronica.

kdesdk applicazioni per gli sviluppatori.

kdetoys applicazioni minori di KDE.

kdeutils utilità varie (visualizzatori di testo semplice, formattatori di floppy ecc.).

kdevelop un IDE (Integrated Development Environment) per sviluppare applicazioni per KDE (quindi in QT3).

kdewebdev contiene strumenti di sviluppo per sviluppare pagine HTML. Il principale è l'editor Quanta Plus.

koffice la suite di applicazioni per ufficio (videoscrittura, foglio elettronico, presentazioni e altri programmi).

qt strumento di sviluppo per sviluppare applicazioni con interfaccia grafica in C++.

Serie di pacchetti t (compresi 101,3 MB, decompressi 245,2 MB) contiene Tetex, una completa distribuzione di programmi, utilità e font atti alla fotocomposizione per elaborare documenti \TeX o \LaTeX .

Serie di pacchetti tcl (compresi 6,3 MB, decompressi 15,6 MB) contiene il supporto per i linguaggi di scripting Tcl/Tk/TclX.

Serie di pacchetti xap, seconda parte contiene altre applicazioni che girano sotto il server grafico. Precisamente, include il browser SeaMonkey (che rimpiazza il vecchio Mozilla), un client per chattare (Gaim), e infine un programma di disegno e fotoritocco (Gimp).

Pacchetti della directory /extra del CD 3

La directory /extra (seconda parte) contiene una serie di programmi aggiuntivi o alternativi a quelli già presenti nella distribuzione, che integrano quelli presenti nell'omonima directory del CD 2.

Pacchetti della directory /kdei del CD 3

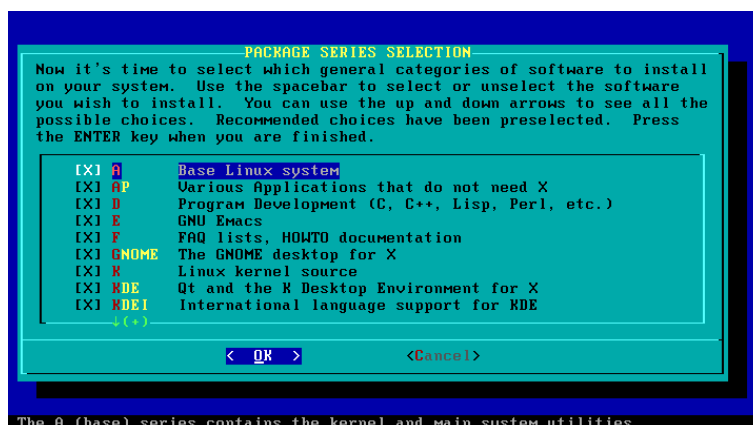
La directory /kdei contiene l'insieme dei pacchetti kdei (compresi 339,3 MB, decompressi 705,5 MB), e cioè i pacchetti col supporto linguistico di KDE per una novantina di lingue diverse.

Pacchetti della directory /testing del CD 3

La directory /testing contiene una serie di programmi considerati non perfettamente stabili o sicuri ma di indubbio rilievo, oppure (com'è il caso di cups o di iptables) di versioni recenti e non ancora sufficientemente testate di programmi presenti nella distribuzione.

O si hanno già le idee molto chiare su ciò che ci serve, oppure – è il caso dei principianti – è meglio abbondare. Potremo in seguito decidere, a ragion veduta, che cosa ci interessa e che cosa no, e rimuovere eventuali pacchetti inutili. Scegliere troppo poco potrebbe originare problemi e malfunzionamenti.

Innanzitutto dobbiamo selezionare i gruppi di pacchetti. Alla voce «PACKAGE SERIES SELECTION» selezioniamo tutto con la barra spaziatrice, ad eccezione della voce KDEI, altrimenti il programma installerà tutti i pacchetti linguistici, che occupano molto spazio e la cui utilità è quanto meno dubbia. Installeremo in un secondo momento, al primo riavvio, il solo pacchetto di KDEI per la lingua italiana:



Alla voce «SELECT PROMPTING MODE» optiamo per «full».



Slackware supporta sei tipi diversi di installazione:

FULL installa tutti i pacchetti senza porre alcuna richiesta di scelta o di conferma.

NEWBIE installa automaticamente tutti i pacchetti indispensabili e chiede se installare o meno, uno a uno, quelli opzionali, fornendone una breve descrizione e indicando lo spazio che occupano. L'opzione **Skip** passa al pacchetto successivo.

MENU simile a **NEWBIE**, non mostra i pacchetti indispensabili che installa automaticamente e presenta un menu unico con tutti i pacchetti opzionali, senza descriverli uno alla volta.

EXPERT installa a richiesta i singoli pacchetti, esigendo continue scelte da compiere mediante un menu interattivo. Conferisce un controllo totale sull'installazione, e consente anche di escludere i pacchetti essenziali (ed è perciò vivamente sconsigliata agli inesperti, in quanto potrebbe risultarne un sistema corrotto o non funzionante).

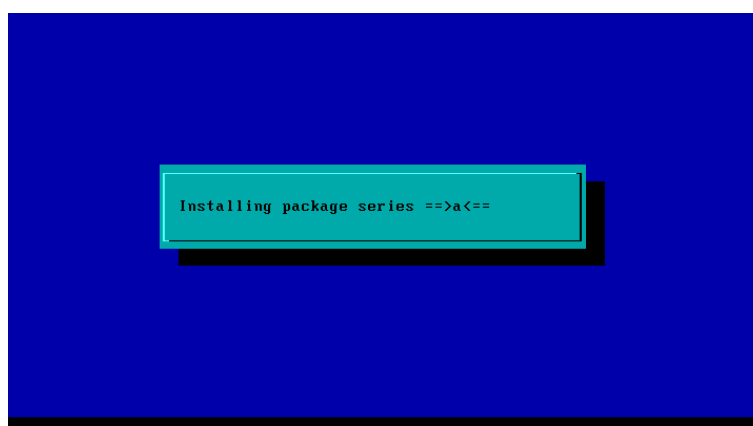
CUSTOM adotta dei tagfiles (particolari file che dicono al programma setup quali pacchetti devono essere installati) creati dall'utente nella directory corrente dei pacchetti.

TAGPATH adotta dei tagfiles nelle subdirectory di un percorso definito dall'utente.

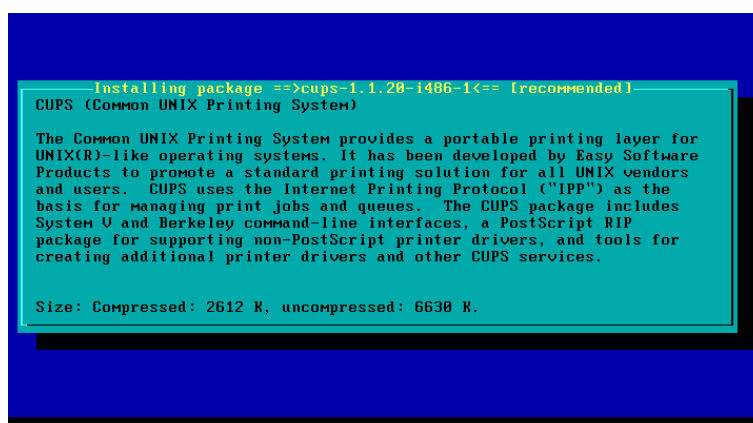
Queste due ultime modalità sono utili per installare velocemente Slackware su più macchine, e di conseguenza destinate a utenti avanzati.

Pertanto, dei vari sistemi di installazione proposti, la soluzione gordiana è la prima, FULL, che installa tutto senza mettere il neofita in difficoltà.

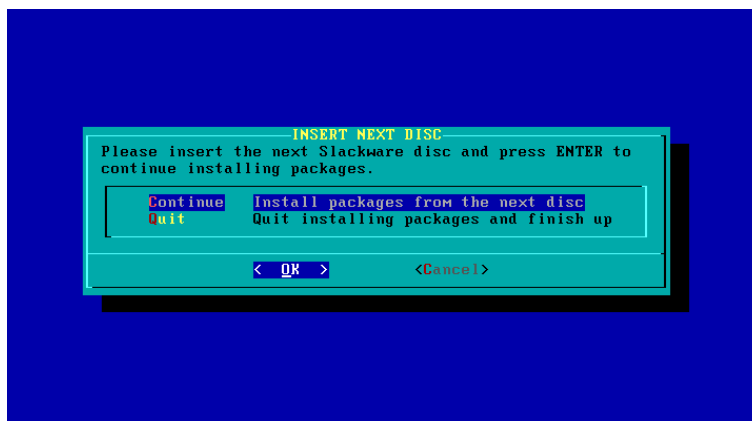
Inizia quindi la decompressione e la copia dei file, che avverte dei pacchetti organizzati per lettere:



e quindi delle caratteristiche dei singoli pacchetti in via di installazione:



A un certo punto il programma ci chiederà di inserire anche il successivo CD, e così fino all'esaurimento dell'installazione:



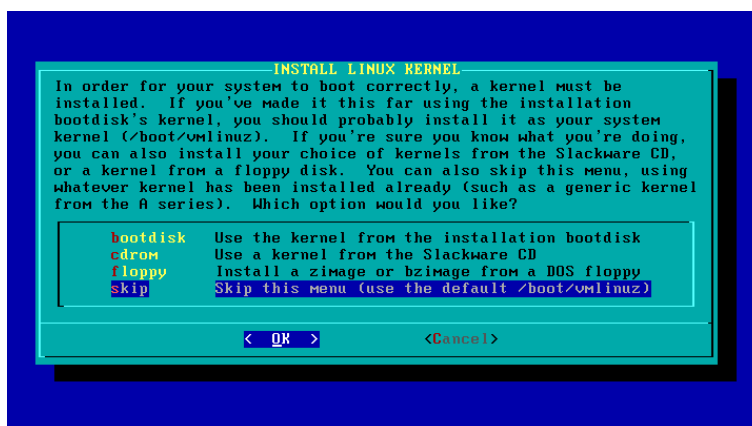
Quindi si avvia la configurazione automatica del sistema. Per una visione d'insieme della destinazione dei vari pacchetti e, più in generale, della struttura del file system di Slackware (che presenta delle peculiarità rispetto ad altre distribuzioni GNU/Linux), vedi [Scheda n. 9 – Le directory di Slackware 11.0](#).

3.8 Refinire l'installazione

Terminata questa fase, ci sono alcune altre cruciali operazioni da effettuare. Il programma di installazione ci pone infatti una serie di questioni atte a configurare propriamente alcuni aspetti fondamentali del sistema.

3.8.1 Scegliere un kernel

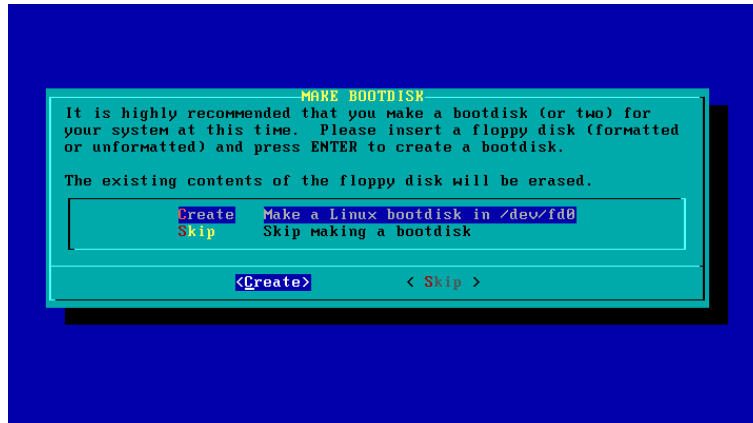
Possiamo scegliere un kernel, per avviare il sistema, diverso da quello usato nell'installazione:



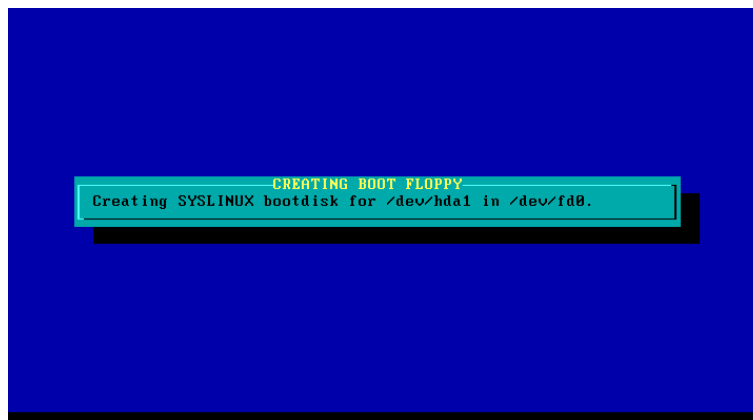
Posto che `bare.i` abbia tutte le caratteristiche che ci servono, saltiamo la scelta selezionando «Skip this menu» e verrà usato per l'appunto `bare.i`. Se invece abbiamo una diversa necessità, ad esempio se il nostro disco fisso è SATA, sceglieremo di installare un diverso kernel (nel caso specifico `sata.i`) dal CD. Se infine avremo scelto di installare il kernel 2.6 da CD o DVD, allora sceglieremo questo. Poiché i kernel sono contenuti nel CD 1, nel primo caso sarà necessario cambiare il CD nel lettore.

3.8.2 Creare un floppy di avvio

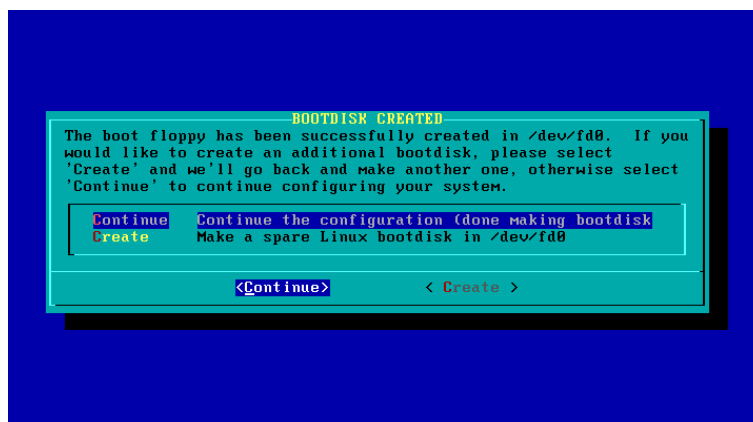
Possiamo creare un floppy di avvio, operazione in genere opportuna:



Di norma il drive viene rilevato automaticamente, e corrisponde al dispositivo `/dev/fd0`:



Una volta creato il dischetto, è possibile crearne un secondo per sicurezza, o passare alla fase successiva dell'installazione:



Nel caso avessimo dei problemi a far partire il programma dal disco fisso, saremo in grado di ricorrere al floppy, anche se in realtà è sempre possibile far partire il sistema dal CD1 di Slackware. Per evitare spiacevoli sorprese, è opportuno controllare il buon funzionamento del floppy provando ad avviare da questo. Inoltre, il floppy può essere creato in qualsiasi momento con l'utilità **pkgtool** oppure col comando: `makebootdisk`.

Scheda n. 7

La gestione dei floppy

Come in tutti i sistemi GNU/Linux, anche in Slackware la gestione dei floppy è riservata a root. Qui sono indicati i comandi fondamentali e più usati.

Formattare un floppy

a) Per effettuare una formattazione di basso livello, dare il comando: `fdformat /dev/fd0[H1440]`. L'opzione `-n` disabilita la verifica, che altrimenti viene effettuata di default.

b) Per creare il file system EXT2 dare il comando: `mkfs.ext2 /dev/fd0`.

c) Per creare il file system FAT dare il comando: `mkfs -t msdos -c /dev/fd0`. Anche qui l'opzione `-c` verifica la correttezza della formattazione.

In tutti questi casi è inoltre sempre possibile aggiungere il parametro `H1440` per specificare la densità del floppy, con la seguente sintassi: `fdformat /dev/fd0H1440`.

d) Per formattare un floppy con capienze superiori, si ricorre all'utilità `superformat`. I formati possibili sono i seguenti, ciascuno seguito dal relativo comando:

1743 KB = `superformat -f /dev/fd0 sect=21 cyl=83`

1840 KB = `superformat -f /dev/fd0 tracksize=23b mss 2rr ssize=2KB`

1992 KB = `superformat -f /dev/fd0 tracksize=12KB cyl=83 mss`

Copiare un floppy

Il metodo più semplice e veloce per copiare direttamente un floppy è ricorrere al comando `diskcopy`.

Un altro metodo nel creare nel disco fisso una immagine del floppy di origine e nel trasferirla poi nel floppy di destinazione. Questi sono i passaggi:

a) creare nella directory `/root` l'immagine col comando: `dd if=/dev/fd0 of=/floppy.img`

b) inserire il floppy di destinazione

c) copiare l'immagine nel floppy di destinazione col comando: `dd if=/floppy.img of=/dev/fd0 [conv=sync]`

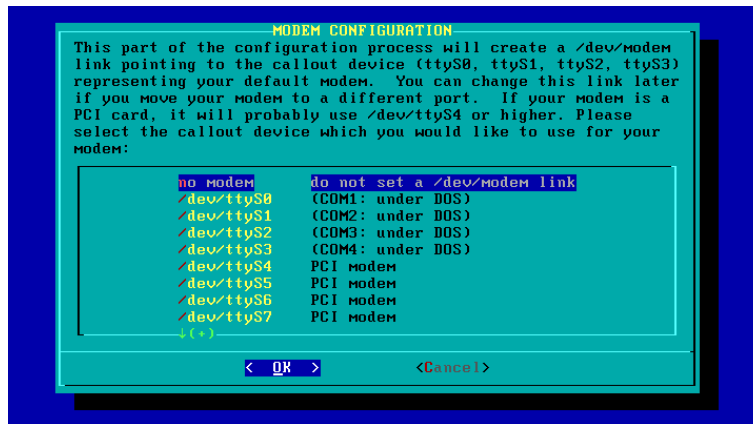
e) rimuovere il file di immagine `*.img` nella directory `/root`.

Poiché a volte può generarsi un errore di I/O se l'ultimo settore del floppy non è tutto scritto, il processo non va a buon fine. Per evitarlo, basta aggiungere al comando descritto al punto c) l'opzione `conv=sync`.

Tutte queste procedure sovrascrivono direttamente il floppy, senza avvertire se esso contiene dei dati; è perciò opportuno controllare se il disco di destinazione può essere, eventualmente, sovrascritto.

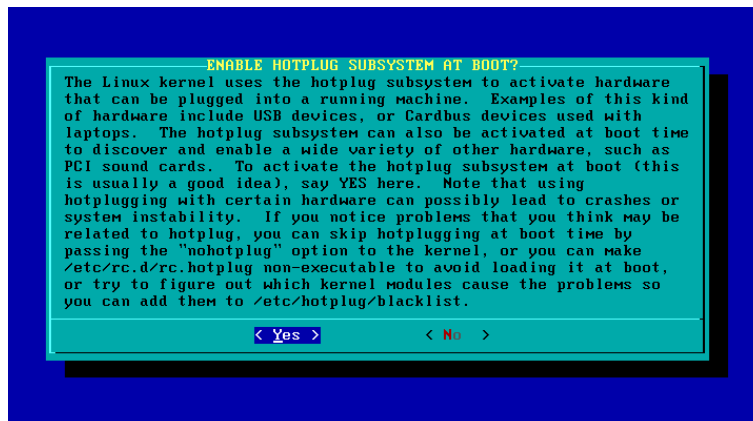
3.8.3 Configurare il modem

Possiamo configurare, se presente, il modem. Ovviamente, i passaggi ulteriori dipendono dal tipo di modem che possediamo, dalle sue caratteristiche e dalla porta in cui è inserito:



3.8.4 Abilitare Hotplug

Possiamo abilitare HOTPLUG all'avvio. È una scelta da fare senz'altro se si usa il kernel 2.4.33.3, perché hotplug si occupa di individuare e gestire le periferiche plug&play:



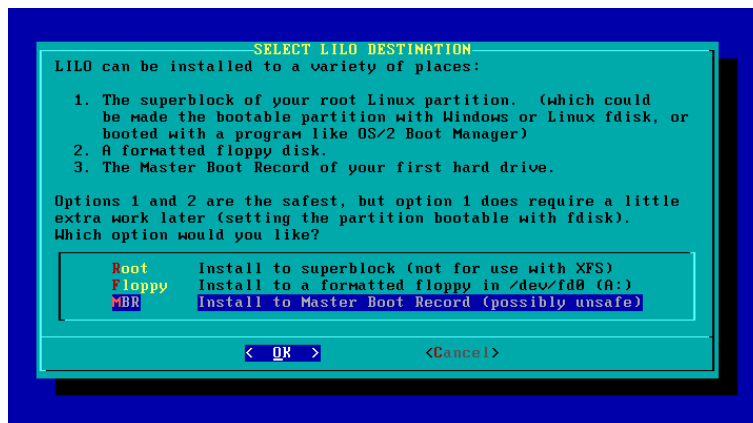
Se invece abbiamo installato un kernel della serie 2.6, possiamo rinunciare a hotplug e passare la gestione delle periferiche a udev.

3.8.5 Installare Lilo

Ora dobbiamo installare Lilo, il bootloader o caricatore di boot (Lilo è l'acronimo di «Linux LOader»), e cioè il programma che si occupa di farci scegliere all'avvio il sistema operativo da montare, Windows, Slackware o eventuali altri sistemi. In questa fase è meglio affidarsi all'installazione automatica, mentre modificheremo Lilo in un secondo momento, per meglio piegarlo alle nostre necessità:



Delle tre opzioni proposte (ROOT, FLOPPY, MBR), scegliamo di installare Lilo sull'MBR (Master Boot Record), che sovrascrive il bootloader di Windows.



L'installazione sulla directory di root richiede che la partizione sia resa avviabile, cosa che si può fare con `fdisk`. L'installazione su floppy richiede naturalmente il suo inserimento nell'apposita unità ogni qualvolta si debba lanciare Slackware.

L'avviso secondo cui l'installazione di Lilo nell'MBR è «possibly unsafe» non deve preoccupare. Ciò si verifica solo in casi rari e particolari: se il BIOS non supporta la capacità del disco e sono installati software come MaxBlast o EZBios, oppure se è già installato un altro bootloader come System Commander o Boot Magic. Con i computer più recenti e le versioni correnti dei programmi, tuttavia, non c'è ragione di temere che l'installazione di Lilo nell'MBR possa causare danni o non avviare regolarmente il sistema.

È importante ricordare alcune cose:

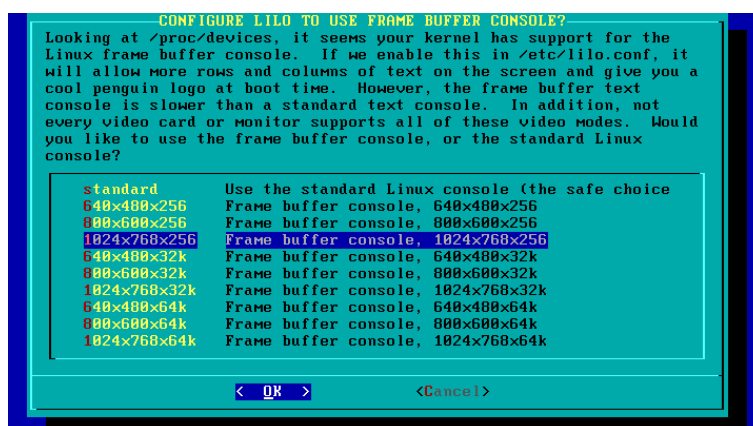
a) il bootloader di GNU/Linux ci permette di scegliere anche altri sistemi operativi, come appunto Windows. Al contrario, il bootloader di Windows non ce lo permette, anche se esiste un modo per modificarlo in questo senso; esiste un'utilità che effettua automaticamente l'aggiunta di ulteriori sistemi operativi OS/2 o GNU/Linux da far partire col bootloader di Windows, e cioè BootPart (reperibile all'indirizzo: <http://www.winimage.com/bootpart.htm>).

b) Il bootloader di Windows viene così sovrascritto, ma se poi decidessimo di disinstallare Slackware, per eliminare anche Lilo e avviare Windows sarà sufficiente ripartire con un floppy di avvio di Windows XP e lanciare il comando `fdisk /mbr` (dalla console di ripristino di Windows 2000 va invece lanciato `fixmbr`), oppure, prima di eliminare Slackware, cancellare Lilo da questa col comando `lilo -u` oppure `lilo -U` (il primo comando ripristina il settore di boot salvato in `/boot/boot.nnnn` dopo averlo

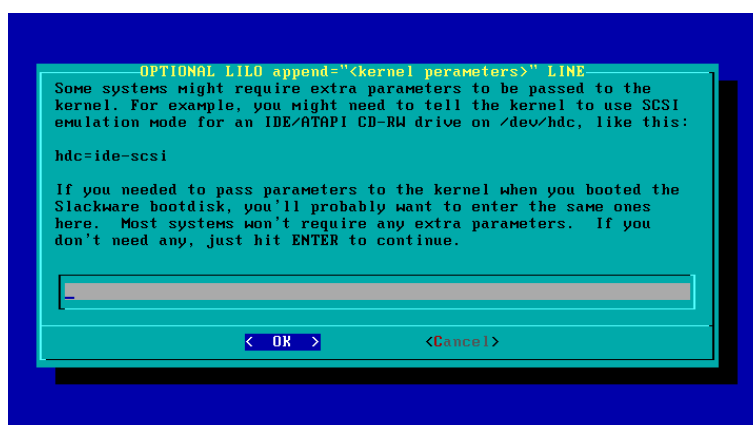
convalidato in base a un *timestamp*, cioè l'indicazione di una data e di un'ora, il secondo si limita al puro e semplice ripristino). Windows ripartirà senza problemi.

c) Se abbiamo già installato Windows nel sistema, la procedura è del tutto lineare: Lilo sovrascriverà il bootloader di Windows. Se invece si installa Windows dopo Slackware, il suo bootloader sovrascriverà Lilo. Di conseguenza sarà necessario entrare in Slackware col floppy di avvio oppure col CD 1 o col DVD e poi da lì reinstallare Lilo.

Se vogliamo migliorare la qualità grafica delle scritte di avvio con un carattere più piccolo e luminoso (e poter vedere anche la simpatica immagine del pinguino in alto a sinistra sullo schermo), dobbiamo attivare il frame buffer per la console:



Infine c'è la possibilità di passare a Lilo alcune righe opzionali («append») per attivare funzioni od opzioni particolari, come l'emulazione SCSI:



3.8.6 Configurare il mouse

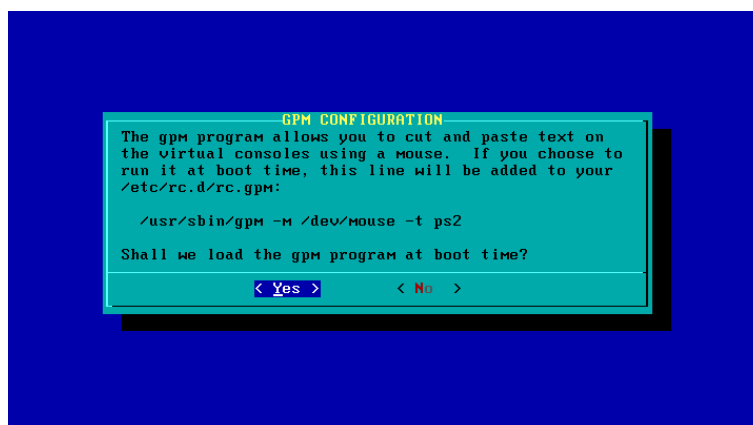
Possiamo configurare il mouse, scegliendo quello in nostro possesso. Slackware ci propone ben venti opzioni, a seconda del mouse che possediamo:



Nel prosieguo considereremo il caso di un mouse PS/2 (il tipo ancor oggi maggiormente diffuso e che non crea problemi con GNU/Linux).

3.8.7 Abilitare GPM

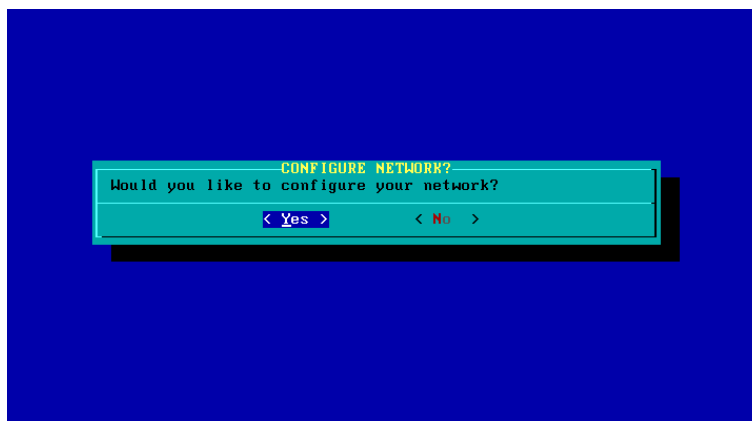
Possiamo abilitare GPM (abbreviazione di «General Purpose Mouse» server), anche in questo caso scelta conveniente:



GPM rende possibile l'uso del mouse all'interno della console (la «finestra di comando» di GNU/Linux), cosicché potremo scorrere il testo con la rotellina del mouse o fare dei comodi copia/incolla altrimenti impossibili (per inserire il testo copiato si preme la rotellina del mouse nel punto di inserimento; nei mouse privi di rotellina, questa funzione può essere emulata premendo contemporaneamente i due tasti).

3.8.8 Configurare la rete

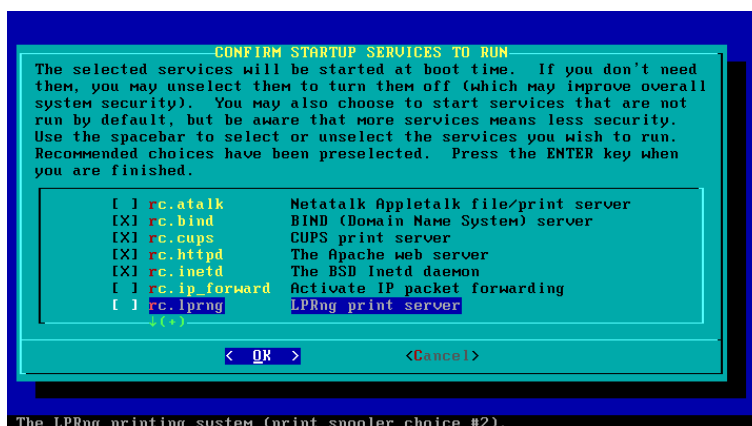
È questo il momento per configurare la rete, utile non solo se abbiamo per l'appunto una rete, ma semplicemente anche se navighiamo con una scheda di rete e un router oppure con un modem seriale:



Rimandiamo la trattazione a un apposito capitolo.

3.8.9 Impostare i demoni

Impostiamo adesso i programmi che Slackware avvia automaticamente (*startup services*) e che lavorano in background, chiamati «demoni». È naturalmente possibile impostare anche in un momento successivo il loro avvio automatico, o mettendo mano ai file di configurazione o mediante l'utilità `pkgtool`:



Si tratta dei seguenti servizi:

Atalk il server per reti Macintosh. Serve dunque solo in un caso specifico e particolare.

Bind Berkeley Internet Name Domain. Si tratta di un DNS server (Domain Name Server), grazie al quale è possibile identificare un determinato computer mediante un nome anziché mediante l'indirizzo numerico delle risorse in rete, e cioè l'IP.

Cups il principale gestore delle stampanti (Common UNIX Printing System). Poiché supporta un vastissimo numero di stampanti ed è di facile configurazione, è meglio preferirlo a `lprng` (attualmente CUPS è il gestore di default in Slackware). Va perciò attivato. Esso è raggiungibile per tutte le operazioni di configurazione da un browser qualsiasi all'indirizzo IP: `http://localhost:631`.

Dnsmasq il server DHCP/DNS, che fornisce questi servizi a una LAN, consentendo l'allocazione dinamica della configurazione di rete dei client.

Httpd il web server Apache. Vi si accede aprendo un browser e digitando l'indirizzo IP: `http://127.0.0.1`, dove è possibile rinvenire anche la documentazione. Questo servizio serve a coloro che vogliono fare del loro computer un server accessibile dall'esterno. In questo caso è necessario o avere un IP statico, oppure fruire di servizi come no-ip (all'indirizzo `http://www.no-ip.com` oppure `http://www.dynu.com`). Ma si tratta di una funzionalità che difficilmente interessa un principiante.

Inetd il BSD Inet Daemon, che serve a far funzionare servizi come `time`, `ftp`, `talk`, `finger` e altri ancora. Esso ascolta sulle porte specificate nel suo file di configurazione e fa avviare il relativo servizio nel momento in cui ne viene fatta richiesta. Per questo è chiamato anche «superdemone». Esso ottimizza le risorse di sistema, avviando il programma che gestisce un determinato servizio solo quando viene effettivamente richiesto. La sua configurazione precisa può essere effettuata in un secondo momento (ad esempio si possono chiudere le porte non in uso, oppure può essere attivato solo per determinati servizi a basso e occasionale traffico). C'è chi sostiene che valga comunque la pena di abilitarlo, chi invece lo sconsiglia, perché tale servizio tiene aperte tutte le porte del computer, e ciò potrebbe rappresentare un rischio per la sicurezza.

Ip_forward serve se il computer deve fungere da router. Normalmente è un servizio che non viene utilizzato, se non per esigenze particolari.

MySQL il server di database offerto da Slackware, potente e versatile. Non è questa la sede per descrivere il funzionamento di MySQL. Se però intendiamo servircene, selezioniamo questa opzione. Non c'è da preoccuparsi se poi, al caricamento del programma, riceveremo un messaggio che, dopo aver cercato di avviarlo, è costretto a chiuderlo, avvertendoci con un laconico «mysql ended». In effetti il programma, se non trova già settati i suoi file di configurazione, non riesce a partire. Sarà un compito successivo imparare a usarlo.

Pcmcia supporta le schede per i laptop.

RPC RPC portmapper demon. Serve, fra le altre cose, a gestire NFS, ossia per usare il Network file system qualora sia nostra intenzione condividere file con altri computer aventi GNU/Linux come sistema operativo.

Samba il gestore delle reti tra GNU/Linux e Windows od OS/2. Serve appunto condividere file e stampanti fra computer aventi GNU/Linux come sistema operativo e altri aventi Windows od OS/2.

Saslauth server di autenticazione basato sulla libreria SASL, usato sovente nei server di posta elettronica.

Scanluns effettua un'analisi (*scannig*) dei LUNs (acronimo di «Logical Unit Numbers») alla ricerca di dispositivi USB.

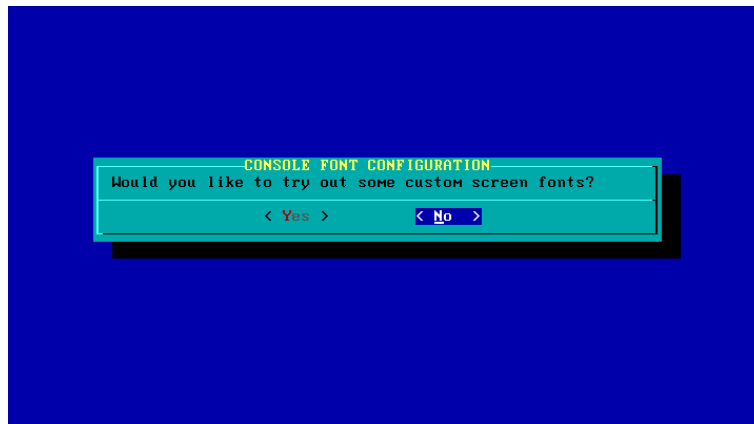
Sendmail il server per gestire la posta elettronica. Se non abbiamo questa esigenza, possiamo sorvolare.

Syslog serve per poter registrare e leggere i messaggi di sistema presenti in `/var/log`.

Sshd consente accessi sicuri e criptati al computer.

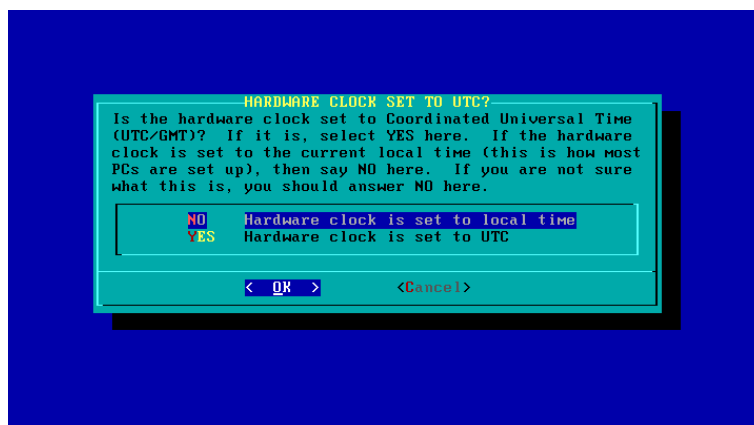
3.8.10 Sostituire i font della console

A questo punto il sistema chiede se vogliamo sostituire i font che il programma usa all'avvio per dare i suoi messaggi mentre si carica. In una prima fase possiamo tranquillamente rispondere di no: ci sarà sempre tempo per cambiarli dopo, se proprio vogliamo:

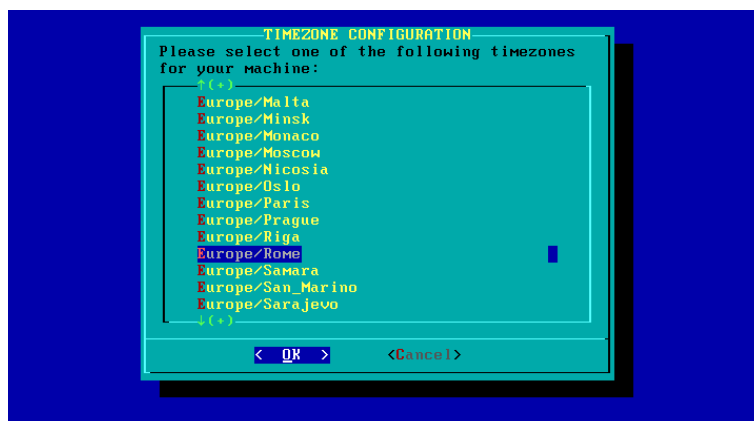


3.8.11 Impostare l'ora di sistema

Impostiamo adesso l'ora di sistema. Le alternative sono UTC e Local Time:



Scegliamo Local time, e andiamo con **PgDown** alla voce **Europe/Rome**:



In ogni caso, UTC (acronimo di Tempo Universale Coordinato) può essere attivato sui computer che montano solo GNU/Linux; se invece si hanno nel computer sistemi operativi diversi (come Windows),

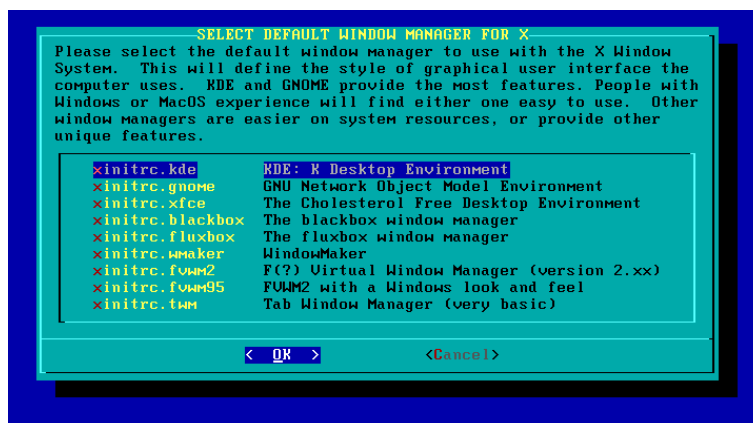
allora è obbligatorio scegliere *Local time*, perché altrimenti, col cambio dell'ora legale, si avrebbero degli errori.

Sarà possibile in seguito modificare le impostazioni dell'ora con l'utilità pseudografica *timeconfig*.

3.8.12 Scegliere un Desktop Environment

GNU/Linux non contiene, nel kernel, programmi atti alla gestione grafica del sistema, e si affida perciò a un programma esterno, lo X Window System; attualmente Slackware ricorre a Xorg, dopo aver abbandonato Xfree dalla versione 10.0. Lo X Window System fornisce tutte le funzioni grafiche per lo schermo, ma a sua volta necessita di un programma che, sfruttando queste funzioni, gestisca le finestre sullo schermo per l'utente: questo programma si dice Window Manager (abbreviato in WM). Vi sono WM minimali, che forniscono soltanto il disegno delle finestre, e WM più avanzati, che offrono anche dei menu accessori per lanciare i programmi. Poiché tuttavia una fetta notevole dell'utenza vuole trovarsi già sul desktop icone, pannelli e menu, funzioni che i WM non sono in grado di svolgere, ecco che esistono anche i Desktop Environment (abbreviato in DE) che presentano tutte queste funzioni. In realtà, le correnti imprecisioni terminologiche derivano anche dal fatto che spesso, ormai, le due ultime espressioni sono interscambiabili: un WM come IceWM offre prestazioni da DE, laddove Xfce ha caratteristiche assai simili a IceWM.

Scegliamo ora un Desktop Environment, cioè il sistema grafico da far partire come default:



Sarà comunque sempre possibile passare da un Desktop Environment all'altro mediante il comando *xwmconfig*. Inoltre si potranno poi aggiungere dei Desktop Environment non inclusi in Slackware, come Gnome, Enlightenment e molti altri. Ormai i gestori di finestre in GNU/Linux sono estremamente integrati, e allo stesso tempo coerenti col sistema operativo e molto vari nella grafica. Slackware offre innumerevoli Desktop Environment.

KDE è, con Gnome, il sistema più potente e dotato di strumenti e di opzioni, ma anche il più pesante, nel senso che richiede innumerevoli risorse. Se il computer è dotato di parecchia RAM, si può optare tranquillamente per KDE. KDE presenta in aggiunta, per la configurazione di molti aspetti del sistema, dei propri strumenti, anche se a volte essi sembrano un po' invasivi e non sempre hanno un funzionamento perfetto: è dunque meglio ricorrere ai più tradizionali e collaudati metodi.

XFCE presenta un'interfaccia leggera, ma altamente funzionale. Utilizza le librerie GTK+ e ha un proprio Window Manager. La gestione grafica è altamente personalizzabile e con funzioni avanzate, grazie a un file manager per gestire il sistema, ma priva di icone del desktop. Possiede inoltre un pannello per lanciare le applicazioni, un gestore dei suoni, un modulo per il pager e uno di supporto a Gnome.

Blackbox presenta un'interfaccia leggerissima. Il desktop è vuoto e i programmi si aprono con menu a tendina attivati dal mouse. Non supporta effetti grafici particolari, ma questo lo rende adatto a computer con scarse risorse. Diverse sue caratteristiche sono personalizzabili direttamente dai file di configurazione.

Fluxbox interfaccia grafica scarna, senza pannelli o icone, ma che offre la possibilità di aprire innumerevoli linguette per i vari programmi. È completamente personalizzabile (supporta infatti le DockApp di Window Maker, che gli offrono molti strumenti aggiuntivi) e si gestisce per intero attraverso il mouse. È tuttavia possibile, attraverso applicazioni aggiuntive (come ROX), dotarlo di icone sul desktop e di un file manager. Deriva da BlackBox.

Wmaker interfaccia grafica leggera, ma funzionale e dotata di innumerevoli strumenti. Inoltre è altamente personalizzabile. Usa prevalentemente il mouse, ma possiede un menu per lanciare applicazioni e configurare alcuni aspetti del sistema.

FVXM2 interfaccia grafica molto leggera. I programmi vengono eseguiti senza integrazione grafica, cioè in modo totalmente indipendente dall'interfaccia stessa.

TWM è forse la più leggera delle interfacce grafiche. Anche in questo caso il desktop è vuoto e i programmi si aprono con menu a tendina attivati dal mouse.

Scheda n. 8

Come installare Gnome

Gnome è, con KDE, il Desktop Environment più ricco e sviluppato. Le più recenti versioni (a partire dalla 2.10) presentano importanti nuove funzionalità, soprattutto nell'ambito della grafica. A differenza di KDE, che appare un insieme estremamente unitario di pacchetti, il team di Gnome non sembra tuttavia star perseguendo uno sviluppo parallelo di tutte le applicazioni.

Slackware e Gnome

Dalla versione 10.2 Slackware ha abbandonato Gnome; su questa scelta si è espresso lo stesso Volkerding, che ha denunciato gli enormi problemi di compilazione di Gnome. Nel Changelog del 26 marzo 2005 egli infatti scriveva:

gnome/*: rimosso dalla -current e lasciato al supporto e alla distribuzione della comunità. Non intendo riprendere tutti i motivi che stanno dietro a questa scelta, che però è stata valutata per più di quattro anni. Esistono già dei buoni progetti per dotare Slackware di GNOME per quelli che lo desiderano, e sono più completi di quanto Slackware non abbia fornito in passato. Perciò, a chi cerca GNOME per Slackware -current io raccomando dei pacchetti ben costruiti in questi due progetti, che perseguono una politica di minima interferenza col sistema base di Slackware:
<http://gsb.sf.net>
<http://gware.sf.net>.

C'è anche Dropline, naturalmente, che è assai conosciuto. Tuttavia, a causa della sua politica di aggiungere PAM e di sostituire numerosi pacchetti di sistema (come l'intero sistema X11) con la sua versione, non mi sento di dare lo stesso assenso a Dropline. Nondimeno, esso rimane un'altra scelta, e dopo tutto il sistema è il vostro, per cui menziono anche quel progetto:
<http://www.dropline.net/gnome>.

Per favore, non si interpreti tutto ciò come una sottovalutazione di GNOME in sé, che (sebbene di solito richieda di essere ulteriormente corretto e risistemato rispetto a com'è fornito, in misura maggiore, diciamo, di KDE o XFce) è una buona scelta per il desktop. Allo stesso modo, di scelte ce ne sono molte altre, ma Slackware non intende fornirle tutte. GNOME è ed è sempre stato una realizzazione in divenire (anche le versioni «stabili» di solito non sono ancora perfezionate), che richiede una squadra per star dietro a tutti i cambiamenti (molti dei quali non sono spesso ben documentati). Io mi auguro davvero che questa scelta migliorerà sia la qualità di Slackware, sia la qualità (e la quantità) delle opzioni di GNOME per essa.

In ogni caso, esistono i tre progetti non ufficiali citati anche da Volkerding per consentire di usare su Slackware questo Desktop Environment.

Freerock Gnome

Il primo progetto è Freerocks Gnome, reperibile all'indirizzo: <http://gsb.freerock.org> e costituito in verità da due filoni: il primo, indicato dalla sigla GSB (acronimo di «Gnome SlackBuild»), è un insieme di SlackBuild atti a compilare i pacchetti di Gnome dai sorgenti. Il secondo, indicato dalla sigla FRG (acronimo di «Freerock Gnome»), consiste nei pacchetti già compilati, pronti per l'installazione. La prima opzione presuppone una certa confidenza con Slackware e con i sistemi di pacchettizzazione. La seconda è offerta in svariate modalità: un installer di rete, il ricorso ai repository di slapt-get, una ISO contenente tutti i pacchetti.

È possibile scaricare il net-installer e lanciarlo mediante lynx (il browser testuale), ovviamente da root e senza il server X attivo, col comando:

```
lynx --source http://gsb.freerock.org/net-install | sh
```

In tal modo di otterrà l'ultima versione stabile. Se invece si vuole avere la current, il comando sarà:

```
lynx --source http://gsb.freerock.org/net-install/current | sh
```

In alternativa si può usare slapt-get, dopo aver aggiunto il repository di Freerocks e la versione che si intende scaricare nel file di configurazione `/etc/slapt-get/slapt-get.rc` (senza dimenticare di dare: `slapt-get --update` dopo l'aggiunta). A questo punto il processo si avvia col comando: `slapt-get --install frgnome`.

Come ultima alternativa, si può scaricare una ISO che contiene già i pacchetti compilati. In questo caso si dovrà seguire la seguente procedura:

1) creare una directory di montaggio col comando: `mkdir /mnt/iso`

2) montare l'immagine col comando: `mount -t iso9660 file.iso /mnt/iso -o loop`

3) andare nella directory dov'è montata l'immagine col comando: `cd /mnt/iso`

4) installare i file col comando: `./frg_install.sh` per un'installazione automatica. Il programma chiede dapprima se si intenda effettuare un'installazione minimale (1) oppure completa (2), e quindi quale versione di alcuni pacchetti (eventualmente dotati di patch specifiche) si voglia installare. Infine chiede l'autorizzazione ad abilitare alcuni script che collocherà in `/etc/rc.d` (e precisamente `rc.howl`, `rc.saslauthd` e `rc.openldap`).

È naturalmente sempre possibile ricorrere a un'installazione interamente manuale, da effettuarsi col comando: `upgradepkg --install-new *.tgz`. Attenzione a non usare semplicemente `installpkg`, oppure ci si ritroverà con diversi pacchetti duplicati (quelli già presenti in Slackware), in due differenti versioni.

Freerock può creare qualche lieve conflitto nel sistema, peraltro risolvibile in maniera molto semplice. Poiché tali problemi mutano con le varie realizzazioni, andrà sempre tenuta sotto controllo la pagina delle FAQ di Freerock, all'indirizzo: <http://gsb.freerock.org/faqs>.

Freerock, nella versione current, è attualmente compatibile con Slackware 10.2 e coi kernel della serie 2.4. Tuttavia alcune funzioni, quali lo Gnome Volume manager, richiedono il kernel 2.6.

Gware

Il secondo progetto è GWARE, reperibile all'indirizzo: <http://gware.sf.net>, che fornisce dei pacchetti non intrusivi per Slackware, cioè dotati della massima compatibilità con la distribuzione. Gware presuppone che sia già presente nel sistema una ventina di pacchetti indispensabili. Inoltre modifica pochi tra quelli già presenti e installa, oltre a quelli di Gnome, alcuni pacchetti richiesti per attivare funzionalità aggiuntive, e cioè `fam`, `howl`, `dbus` e `hal`, che modificano alcuni file di sistema o installano nuovi file di configurazione o di avvio. Infine, richiede la modifica manuale di un file per far funzionare correttamente `udev`.

Prima dell'installazione, è pertanto necessario leggere attentamente il documento sui requisiti, presente all'indirizzo: <ftp://ftp.gware.org/10.2/2.12.0/packages/INSTALL>.

I passaggi per l'installazione sono i seguenti:

1) si scarica uno script che automatizza tutte le operazioni. Per la revisione G (allo stato attuale, la più recente), lo script prende il nome di `get-gware-G.sh`; ricorrendo a `wget`, si dà il comando:

```
wget ftp://gware.org/10.2/2.12.0/packages/get-gware-G.sh
```

2) si rende eseguibile lo script col comando: `chmod +x get-gware-G.sh`

3) si avvia lo scaricamento e l'installazione dei pacchetti col comando: `./get-gware-G.sh`

Gware sostituisce alcuni pacchetti con la sua propria versione, modifica `/etc/inetd.conf` e `/etc/rc.d/rc.local` e aggiunge alcuni file di configurazione (`/etc/rc.d/rc.howl`, `/etc/rc.d/messagebus`, `/etc/rc.d/rc.hald`).

Anche Gware – attualmente compatibile con Slackware 10.2 – può creare qualche lieve problema nel sistema, peraltro risolvibile in maniera molto semplice. Poiché tali problemi mutano con le varie realizzazioni, andrà sempre tenuta sotto controllo la pagina delle FAQ di Gware, all'indirizzo: <http://www.gware.org/faq.php>.

Dropline

Il terzo progetto è Dropline, reperibile all'indirizzo: <http://www.dropline.net/gnome>; esso si differenzia dai primi due per seguire una politica relativamente autonoma da Slackware, ad esempio fornendo proprie versioni modificate del sistema X11 e adottando pacchetti per la sicurezza, come PAM, estranei alla distribuzione. Si presenta pertanto come piuttosto intrusivo nel sistema. Va ricordato che Dropline Gnome è ottimizzato per i686, gira col kernel di default di Slackware, su cui viene compilato, ma richiede il kernel 2.6 per attivare alcune funzioni come il montaggio automatico dei dispositivi. Infine, può non funzionare correttamente su versioni più vecchie di Slackware e sulla current.

Innanzitutto è necessario scaricare dal sito di Dropline Gnome l'installer, un pacchetto nel classico formato `.tgz`, da installare con `installpkg`. Si lancia quindi il comando `dropline-installer` e a questo punto, dal menu, si scelgono le operazioni da effettuare. Il menu presenta alcune voci, tutte facilmente comprensibili:

Installa = installa una copia nuova e integrale di Dropline-Gnome.

Aggiorna = aggiorna un'installazione precedente coi pacchetti più recenti.

Vedi aggiornamenti = si limita a mostrare l'elenco dei pacchetti disponibili per un eventuale aggiornamento.

Solo download = scarica i pacchetti disponibili senza installarli. Essi vengono resi reperibili di default nella directory `/var/cache/dropline-installer` per una successiva installazione manuale. In questo caso è conveniente installare dapprima il pacchetto Orbit, atto ad aggiornare il database dei pacchetti di Gnome presenti nel sistema.

Rimuovi = disinstalla Dropline-Gnome.

Preferences = consente di settare alcuni parametri di installazione:

Impostare sorgente = consente di scegliere il supporto dei sorgenti (rete, CD-Rom, disco fisso).

Impostare proxy = consente di specificare un proxy HTTP.

Impostare `directory` = consente di scegliere una directory diversa da quella di default per conservare i pacchetti scaricati.

Keep old Programs = mantiene i pacchetti vecchi o li elimina una volta scaricati quelli più recenti.

Keep Install Files = mantiene i vecchi file di installazione o li elimina una volta effettuati gli aggiornamenti.

Esci = esce dal menu.

Il programma di installazione costruisce una lista dei pacchetti, organizzati in categorie. Ne citiamo i più significativi:

Application Development = programmi di sviluppo come Pkgconfig, Intltool e Glade, e inoltre della documentazione.

Desktop Extras = l'editor per gconf, alcuni giochi, Ghex (editor esadecimale).

GNOME Desktop = la piattaforma Gnome essenziale.

Internet Software = Firefox, Thunderbird, Drivel, Epifany, Gaim, GnomeMeeting, Xchat.

Media Software = Totem, Rhythmbox, Eince, Gthumb0, Gimp, Gtkam, Inkscape.

Office Applications = la suite Office di Gnome, Evolution (in preparazione anche OpenOffice).

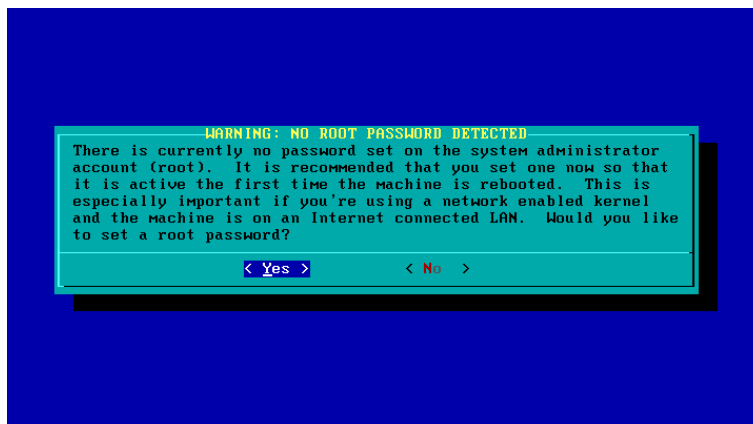
System Software = pacchetti di sistema indispensabili (librerie ecc.) e i pacchetti del server X.

È possibile scegliere le categorie nel loro insieme o scegliere pacchetto per pacchetto. Se si opta per l'installazione per categorie, l'installer offre una ulteriore scelta: un'installazione completa (di quasi 250 pacchetti), un'installazione selettiva e una minimale.

Infine, è disponibile anche la ISO per creare un CD in cui si troveranno tutti i pacchetti.

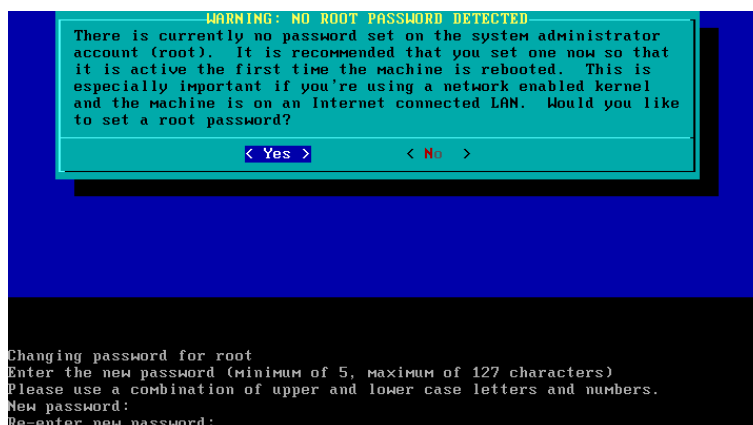
3.8.13 Impostare la password di root

Ci viene richiesta una password di root:



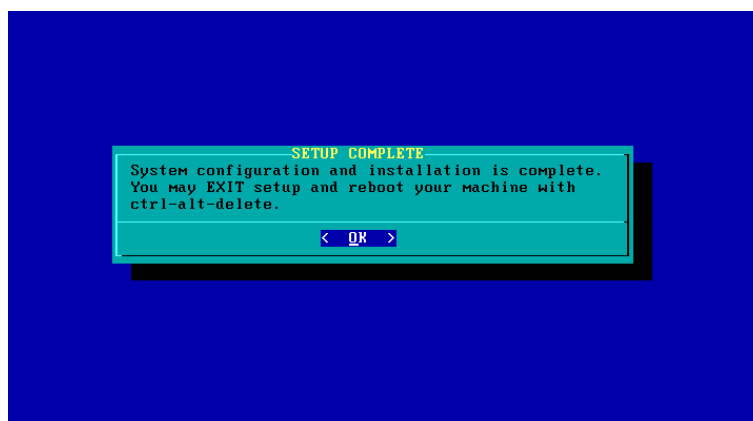
È opportuno scegliere, come sempre, una stringa non troppo semplice od ovvia, consistente in una sequenza di lettere maiuscole e minuscole, numeri e simboli alfanumerici. Se risulta troppo semplice (ad esempio breve e composta da sole lettere minuscole o, peggio, consistente in una parola del dizionario), è il programma stesso a non accettarla. Impostare una password non sembra strettamente necessario, perché è ovviamente possibile accedere al sistema anche senza; se poi noi siamo l'unico utente del nostro computer, la cosa potrebbe ritenersi del tutto trascurabile. In realtà, la configurazione di alcuni programmi (come CUPS o Samba mediante SWAT) richiede obbligatoriamente una password (non si riesce ad entrare in CUPS o in SWAT senza password anche se si è già root).

Una volta digitata la password, se essa viene accettata, dobbiamo digitarla una seconda volta per sicurezza:

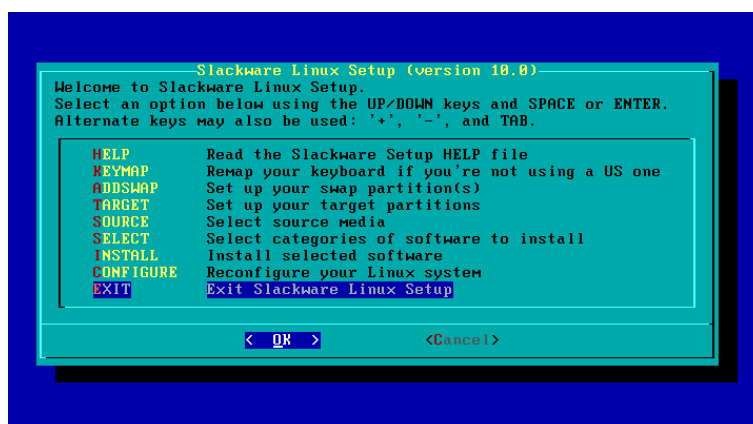


3.8.14 Riavviare il sistema

A questo punto, non resta che aspettare il completamento dell'installazione:



e infine riavviare il sistema con **Ctrl+Alt+Canc**, dopo aver tolto il CD dal lettore:



3.9 Installare Slackware da floppy

Se il computer non supporta il bootstrap da CD, com'è per molte macchine datate, l'installazione da floppy non comporta alcun problema. È sufficiente creare tre floppy copiando i file pertinenti dal CD 1 di Slackware, e precisamente dalla directory `/bootdisks` copiare il kernel `bare.i` (se è questo il kernel che ci serve) e dalla directory `/rootdisks` copiare i file `install.1` e `install.2`. Il procedimento può essere avviato da Windows con l'utilità Rawrite, oppure da Slackware da console con la seguente procedura:

- a) per creare il BOOT DISK, da `/bootdisks` lanciare il comando: `cat bare.i > /dev/fd0`
- b) per creare il ROOT DISK 1, da `/rootdisks` lanciare il comando: `cat install1 > /dev/fd0`
- c) per creare il ROOT DISK 2, da `/rootdisks` lanciare il comando: `cat install2 > /dev/fd0`

Sarà poi sufficiente inserire il BOOT DISK e i due ROOT DISK in sequenza quando richiesto, avendo già inserito nel lettore il CD 1 di Slackware per avviare l'installazione.

Scheda n. 9

Le directory di Slackware 11.0

L'organizzazione di Slackware, che rispetta il Linux Filesystem Hierarchy Standard (LFHS), non è troppo dissimile da quella delle altre distribuzioni GNU/Linux; tuttavia non manca di alcune peculiarità. Vediamo com'è articolata la struttura delle principali directory.

/bin contiene i file binari (cioè eseguibili) essenziali per il funzionamento del sistema operativo, accessibili non solo a root, ma a tutti gli utenti. Di solito viene modificata solo a causa di aggiornamenti.

/boot contiene i file di avvio per l'esecuzione del sistema operativo, cioè il kernel e gli altri file di configurazione dell'avvio del sistema.

/cdrom è il punto di montaggio del lettore CD, automaticamente rilevato durante l'installazione.

/dev contiene la mappatura dei dispositivi, cioè i file che corrispondono alle periferiche.

/etc contiene gli script e i file di configurazione del sistema (dai file di configurazione di X Window al database utenti, fino agli script di inizializzazione del sistema) e di numerosi programmi.

/home contiene le directory principali degli utenti e le relative subdirectory (esclusa quella di root).

/lib contiene le librerie del sistema (le più importanti sono le librerie C, il dynamic loader, le librerie ncurses) e, nella subdirectory `/modules`, i moduli del kernel.

/lost+found all'avvio del sistema, viene effettuato un controllo di eventuali errori del file system. Se ne vengono individuati, viene lanciato il programma `fsck` per tentare di correggerli. Le parti corrette del file system vengono scritte per l'appunto nella directory `/lost+found`.

/media contiene i punti di montaggio delle periferiche rimovibili (come CD, DVD, floppy e pendrive).

/mnt contiene i punti di montaggio delle periferiche fisse (come i dischi fissi).

/opt contiene i pacchetti opzionali. Di default, Slackware installa qui anche KDE. Non è cattiva norma installare in questa directory gli ulteriori pacchetti aggiunti dagli utenti.

/proc punto di montaggio per interagire col kernel. Si tratta di un file system virtuale che non risiede realmente sul disco. Contiene informazioni sul kernel e sui processi in esecuzione, conservate sotto forma di file in questa directory. Reciprocamente, è possibile inviare informazioni al kernel attraverso alcuni di questi file.

/root la directory principale (home) dell'utente root. Essa è collocata per l'appunto in /root e non in /home, come per gli altri utenti del sistema. Infatti, nel caso che la home di root fosse una partizione diversa da / e non fosse possibile montarla per qualche motivo, all'utente root sarebbe difficile eseguire il login per risolvere il problema, essendo la sua home directory collocata sul file system danneggiato.

/sbin contiene i comandi del sistema operativo che solo root può lanciare. Se un utente cerca di lanciali dal suo account, riceverà il messaggio: «command not found», poiché /sbin non risiede nel suo path. Per trasformarsi in root e poterli così lanciare, si darà il comando su per restare comunque sempre nella directory di partenza o il comando su - per trovarsi nella home di root. In alternativa, andrà indicato da parte dell'utente il percorso completo dell'eseguibile.

/sys contiene dei file relativi all'hardware di sistema.

/tmp contiene i file temporanei, ma anche una serie di directory di configurazione di alcuni dispositivi. Certe distro, ma non Slackware, la puliscono in modo automatico all'avvio del sistema.

/usr contiene i programmi a disposizione degli utenti. La subdirectory /usr/X11R6 contiene lo X system del sistema (e cioè il server grafico), i cui file di configurazione risiedono invece in /etc/X11.

/var contiene file di log (cioè i resoconti) del sistema e dei file gestiti dai processi di sistema (come la cache dei dati). Questa directory è pertanto utilizzata per conservare tutte quelle informazioni che cambiano di frequente.

Capitolo 4

Avviare Slackware per la prima volta e spegnerla

4.1 Avviare Slackware

4.1.1 Entrare nel sistema

Dopo il riavvio e il caricamento del programma, ci si presenta il prompt, nella forma:

```
Welcome to Linux 2.4.33.3 (tty 2)
```

```
darkstar login:
```

Dobbiamo inserire root come username e la password che abbiamo scelto. Comparirà il prompt nella forma:

```
root@darkstar:~#
```

A questo punto ci siamo loggati come root, siamo davanti alla console e possiamo aggiungere al sistema uno o più utenti, ed effettuare altre operazioni.

4.1.2 Leggere i messaggi di sistema

Possiamo verificare con calma, una riga alla volta, i messaggi che il programma ci invia durante il caricamento, e che passano velocemente sullo schermo, dando il comando: `dmesg | more`, oppure: `dmesg | less` (che consente anche di ritornare indietro con i tasti freccetta o con **PagDown**) alla ricerca di eventuali errori o altro. In alternativa, possiamo andare a leggere i file `/var/log/syslog` e `/var/log/messages`.

4.1.3 Avviare un Window manager

A questo punto, facciamo partire il nostro Desktop Environment col comando `startx` (X è il nome del server grafico), e si caricherà KDE, secondo quanto abbiamo scelto in fase di configurazione.

4.2 Italianizzare Slackware

4.2.1 Impostare la tastiera

Passiamo così alla prima, rapida configurazione di KDE, scegliendo lingua ed effetti. Ricordiamo che il settaggio della tastiera italiana mediante `xorgconfig` non modifica le impostazioni di default di Kde.

Se qualcosa non avesse funzionato nell'impostazione della tastiera da console durante l'installazione, imposteremo in qualsiasi momento la tastiera italiana col comando: `loadkeys it`, che carica il file `/usr/share/kbd/keymaps/i386/qwerty/it.map.gz`. Tale configurazione non vale per KDE.

4.2.2 Italianizzare KDE

Gli utenti di KDE devono italianizzare questo DE. Ciò richiede tre operazioni, la prima relativa alla lingua dell'ambiente grafico, le altre due alla tastiera.

a) Poiché in sede di installazione abbiamo escluso KDEI, dobbiamo adesso installare il pacchetto della lingua italiana per KDE. Montiamo il CD 3 di installazione col comando: `mount -t iso9660 /dev/hd# /mnt/cdrom` (o comunque si chiami il dispositivo: se si è attivata l'emulazione SCSI potrebbe essere qualcosa come `/dev/sg0`), entriamo nella directory `/KDEI` col comando: `cd /mnt/cdrom/slackware/kdei` e installiamo con `installpkg` il pacchetto `kde-i18n-it-3.5.4-noarch-1.tgz`. A questo punto entriamo in **Centro di controllo – Regionali e accesso facilitato – Lingua e Paese** e dalla linguetta **Aggiungi lingua** scegliamo **Italiano**.

b) Quindi provvediamo a settare la tastiera dal **Centro di controllo – Regionali e accesso facilitato – Mappatura della tastiera**, scegliendo quella Internazionale a 105 tasti.

c) Infine, alla voce **Mappature disponibili**, aggiungiamo quella italiana, facendo in modo che risulti la prima dell'elenco mediante i tasti freccia azzurri in basso a destra.

4.2.3 Italianizzare altre applicazioni

Altre applicazioni, come quelle che girano nativamente sotto Gnome, rimarranno in inglese. Per italianizzare anche queste, dobbiamo introdurre a mano un paio di modifiche in due file di configurazione.

Nel file `/etc/profile.d/lang.sh` commenteremo¹ la riga:

```
export LANG=us_US
```

e aggiungeremo la riga:

```
export LANG=it_IT@euro
```

Nel file `/etc/profile.d/lang.csh` commenteremo la riga:

```
setenv LANG us_US
```

e aggiungeremo la riga:

¹Commentare una riga significa aggiungere, all'inizio, il simbolo del cancelletto (#). In tal modo essa non verrà letta dal programma. Il risultato è lo stesso che cancellarla, ma questa tecnica è migliore perché consente di mantenere delle righe che potrebbero tornare utili in un secondo tempo, e quindi si usa per inserire nei file delle righe esplicative. Ad esempio si avrà:

```
# Questo è un commento
```

Ciò vale almeno per i file di configurazione e per gli script di bash, che è quanto ci interessa, mentre altri linguaggi usano una differente sintassi.


```
setenv LANG it_IT.
```

Ciò è utile anche per italianizzare tutte le applicazioni che utilizzano il sistema di internazionalizzazione gettext (e che sono veramente numerose, compresi molti dei comandi di sistema), che altrimenti risulterebbero nella lingua originale (e cioè il più delle volte in inglese).

4.3 Spegnere Slackware

4.3.1 Uscire da una sessione

Uscire dal server X non significa affatto essere usciti da una sessione. La console è infatti ancora attiva per l'utente che vi è acceduto. Se l'accesso ha avuto luogo da root, ciò può introdurre una grave falla nella sicurezza, poiché a quel punto è possibile continuare a svolgere qualsiasi operazione e aprire una sessione di qualunque utente senza digitare la sua password. Per chiudere realmente la sessione da console si adotta la combinazione di tasti **Ctrl+d**, oppure le stringhe `exit` o ancora `logout`.

4.3.2 Spegnere il computer da root

Di default, solo l'utente root può spegnere il computer. I comandi classici e più sicuri sono: `poweroff`, che in realtà è un link a `halt`, e `shutdown -h now`, che ordinano al sistema di uscire da Slackware istantaneamente e di spegnersi.

Se invece dell'opzione `now` si passa `+n`, dove `n` è un numero determinato, il sistema si chiuderà dopo `n` minuti. In questo caso tutti i processi attivi sono informati del prossimo spegnimento, dando modo ad alcuni programmi di richiedere il salvataggio di eventuali file aperti. Ad esempio, il comando `shutdown -h +10` fa sì che il sistema si spenga dopo dieci minuti. L'opzione `now` è indicata per sistemi domestici o monoutente, mentre l'opzione `+n` è assai utile in sistemi multiutente, perché avverte tutti del prossimo spegnimento. Il comando `shutdown -h 0` è naturalmente lo stesso di `shutdown -h now`. Se invece dell'opzione `-h` si adotta l'opzione `-r` il sistema viene riavviato. In alternativa, per spegnere il sistema si può dare il comando `halt` e per riavviarlo il comando `reboot`. I comandi `halt` e `reboot` (da tutti i runlevel diversi da 0 e 6) richiamano in realtà `shutdown`, ma sono meno elastici, non possono essere accompagnati da determinazioni temporali e non forniscono alcun avviso del prossimo spegnimento. La pressione dei tasti **Ctrl+Alt+Canc** determina il riavvio del sistema indipendentemente dai privilegi di root per accedere ai comandi. Poiché tuttavia questa combinazione di tasti in realtà richiama il comando `/usr/sbin/ctrlaltdel`, l'amministratore può modificarne i permessi in modo da rendere impossibile questa operazione ai semplici utenti.

4.3.3 Spegnere fisicamente il computer

Col kernel 2.4 Slackware termina il suo lavoro, ma il computer rimane acceso. Se si desidera spegnerlo automaticamente, è necessario editare il file `/etc/rc.d/rc.modules` decommentando nella sezione APM support la riga:

```
/sbin/modprobe apm
```

Col kernel 2.6, se il supporto ACPI o APM è abilitato nel kernel stesso, non serve effettuare alcun intervento: il computer si spegne da solo.

4.3.4 Spegnerne il computer da utente

Se si è loggati come utente, si deve terminare la sessione, passare a root col comando `su [-]`, inserire la password di root e quindi lanciare il comando di spegnimento.

Perché anche gli utenti possano spegnere il computer è necessario usare l'utilità **sudo** (abbreviazione di «superuser do»). Il file è già configurato per consentire a tutti gli utenti due tipi di operazioni. Se decommentate, le righe finali, e precisamente:

```
# Samples
# %users ALL=/sbin/mount /cdrom,/sbin/umount /cdrom
# %users localhost=/sbin/shutdown -h now
```

consentono a tutti gli utenti e senza ricorso alcuno alle password rispettivamente di montare e smontare i CD-ROM senza apportare modifiche a `/etc/fstab` e di spegnere il computer.

Lievemente più complessa appare la procedura per assegnare solo ad alcuni utenti dei permessi particolari. Ipotizziamo di voler consentire a un certo numero di utenti la possibilità di spegnere il computer sia con `shutdown` sia con `poweroff` e di poterlo riavviare con `reboot`. È sufficiente a questo fine aggiungere nel file `/etc/sudoers` alle righe già presenti quelle sotto indicate, in modo che risulti secondo la seguente sintassi:

```
# Host alias specification

# User alias specification
User_Alias NOME_ALIAS = nome_utente, nome_utente, ...

# Cmnd alias specification
Cmnd_Alias NOME_COMANDI = nome_comando, nome_comando, ...

# Defaults specification

# User privilege specification
root ALL=(ALL) ALL

# Uncomment to allow people in group wheel to run all commands
# %wheel ALL=(ALL) ALL

# Same thing without a password
# %wheel ALL=(ALL) NOPASSWD: ALL
NOME_ALIAS ALL = NOPASSWD: NOME_COMANDI

# Samples
# %users ALL=/sbin/mount /cdrom,/sbin/umount /cdrom
# %users localhost=/sbin/shutdown -h now
```

Facciamo un esempio. Per conferire il permesso di spegnere il sistema coi comandi `poweroff` e `shutdown -h now` e di riavviarlo con `reboot` agli utenti `tizio`, `caio` e `sempronio`, scegliamo di conferire a questo gruppo di utenti l'alias `UTENTI`, di raccogliere i comandi nell'alias `SPEGNI` e di consentire loro di svolgere queste operazioni senza inserire la password. Il file `/etc/sudoers` apparirà di conseguenza come segue:

```
# Host alias specification

# User alias specification
User_Alias UTENTI = tizio, caio, sempronio
```

```
# Cmnd alias specification
Cmnd_Alias SPEGNI = /sbin/poweroff, /sbin/shutdown -h now, /sbin/reboot

# Defaults specification

# Runas alias specification

# User privilege specification
root ALL=(ALL) ALL

# Uncomment to allow people in group wheel to run all commands
# %wheel ALL=(ALL) ALL

# Same thing without a password
# %wheel ALL=(ALL) NOPASSWD: ALL
UTENTI ALL = NOPASSWD: SPEGNI

# Samples
# %users ALL=/sbin/mount /cdrom,/sbin/umount /cdrom
# %users localhost=/sbin/shutdown -h now
```

I vari *nome_utente*, raccolti sono l'unico *NOME_ALIAS* sono ovviamente i nomi degli utenti autorizzati a effettuare i comandi presenti in *NOME_COMANDI*, tutti nomi che possono essere scelti a piacere. Eliminando nell'ultima riga la stringa *NOPASSWD:* agli utenti autorizzati sarà richiesta la propria password per effettuare le operazioni. Tali utenti potranno spegnere il computer col comando `shutdown -h now` e riavviarlo col comando `reboot` con la seguente sintassi: `sudo /sbin/shutdown -h now` e `sudo /sbin/reboot`. È necessario inserire l'indirizzo completo, perché il comando risiede nella directory `/sbin`, di solito non presente nel path degli utenti (a meno di non creare appositi link per evitare di digitare il percorso completo). Se poi volessimo abbreviare i comandi, potremmo sempre ricorrere a degli alias. In questo caso sarà necessario creare nella home di ciascun utente il file `.bash_profile` (che di default Slackware non crea) includendovi delle seguenti righe, che definiscono degli alias:

```
alias poweroff='sudo /sbin/poweroff'
alias shutdown='sudo /sbin/shutdown -h now'
alias reboot='sudo /sbin/reboot'
```

Esiste infine la possibilità di conferire i permessi di spegnimento e riavvio a un solo gruppo, che per ipotesi chiamiamo gruppo «spegnimento». A questo fine le righe pertinenti in `/etc/sudoers` dovranno risultare:

```
# Cmnd alias specification
Cmnd_Alias SPEGNI = /sbin/reboot,/sbin/shutdown
%spegnimento ALL = NOPASSWD: SPEGNI
```

Inoltre va creato in `/etc/profile.d` lo script `spegnimento.sh`, che risulterà come segue:

```
if id -Gn | grep -q spegnimento
then
```

```
alias shutdown='sudo /sbin/shutdown'  
alias reboot='sudo /sbin/reboot'  
fi
```

In questo modo verrà assegnata esclusivamente agli utenti del gruppo la possibilità di spegnere il computer, e per essi e solo per essi verranno creati gli alias.

Capitolo 5

Ottimizzare il server grafico

5.1 Funzionamento del server grafico

Il server grafico di Linux è X. Questo viene lanciato col comando `startx`, che invoca `xinitrc`. Se nella home dell'utente non è presente il file `.xinitrc`, allora viene usato `/etc/X11/xinit/xinitrc`.

Questo script cerca i file definiti dall'utente o quelli predefiniti del sistema: dapprima `.Xresources`, `.Xmodmap` ed eventualmente `.Xkbmap` nella home dell'utente, quindi `.Xresources`, `.Xmodmap` ed eventualmente `.Xkbmap` nella directory `/usr/X11R6/lib/X11/xinit` (file che sono in realtà dei link simbolici a `/etc/X11/xdm`). `Xmodmap` e `Xkbmap`, se presenti, vengono usati per configurare la tastiera, mentre `Xresources` viene letto per assegnare valori di default ad applicazioni particolari. Quindi, `xinitrc` tenta di avviare l'ambiente desktop o anche solo un Window Manager di base fra quelli contenuti in `/etc/X11/xinit` eseguendo lo script `.xinitrc` presente nella home, o, se non lo trova, il file predefinito, collocato in `/var/X11R6/lib/xinit/xinitrc` (cioè uno degli script `xinitrc.nome_window_manager`) presenti in `/etc/X11/xinit/xinit`.

Se il sistema effettua l'avvio al runlevel 5 (cioè l'accesso diretto alla modalità grafica), viene lanciata un'applicazione speciale del client X: l'utente deve effettuare l'autenticazione usando il display manager prima che venga lanciato un ambiente desktop oppure un window manager. In Slackware sono disponibili due diversi display manager per la gestione dell'autenticazione utente:

kdm: il display manager di KDE, che abilita l'utente a spegnere, riavviare o a registrarsi nel sistema. È presente in `/opt/kde/bin`.

xdm: un display manager di base che permette all'utente di effettuare solo una registrazione nel sistema, presente in `/usr/X11R6/bin`.

Se poi si installa anche GNOME in una delle sue versioni, verrà aggiunto anche **gdm**: il display manager di GNOME, che – allo stesso modo – abilita l'utente a configurare la lingua, le impostazioni, lo spegnimento, l'avvio o la registrazione nel sistema.

I display manager fanno riferimento al file `/etc/X11/xdm/Xsetup_0` per impostare la schermata di registrazione. Effettuata la registrazione nel sistema, lo script `/etc/X11/xdm/GiveConsole` assegna all'utente la proprietà della console. Successivamente, lo script `/etc/X11/xdm/Xsession` entra in esecuzione per compiere molti dei compiti di cui si occupa in genere lo script `xinitrc` all'avvio di X nel runlevel 3, incluse le impostazioni del sistema e delle risorse utente, oltre all'esecuzione degli script nella directory `/etc/X11/xinit/xinitrc`.

I display manager **kdm** e **gdm** consentono agli utenti di specificare quale ambiente desktop utilizzare durante l'autenticazione scegliendolo dal menu **Sessione** (si può accedere selezionando **Pulsante menu principale** [sul pannello] – **Preferenze** – **Più Preferenze** – **Sessioni**). Se il display manager non specifica l'ambiente desktop, lo script `/etc/X11/xdm/Xsession` controllerà i file `.xsession` e `.Xclients`

nella home directory utente per decidere quale ambiente desktop dovrà essere caricato. Come ultima risorsa viene usato il file `/etc/X11/xinit/Xclients` per selezionare un ambiente desktop oppure un Window Manager da usare allo stesso modo del runlevel 3.

Quando l'utente termina una sessione X sul display predefinito (:0) ed effettua l'uscita, lo script `/etc/X11/xdm/TakeConsole` si avvia e assegna nuovamente la proprietà della console all'utente root. Il display manager originale, che è rimasto operativo dopo la registrazione dell'utente, prende il controllo eseguendo lo spawn di un nuovo display manager. Questo riavvia il server X, visualizza una nuova finestra di login e avvia ancora l'intero processo. L'utente viene riportato al display manager dopo aver effettuato un logging out da X dal runlevel 5.

Il file `/etc/X11/xinit/xinitrc` contiene, nella prima parte, alcune impostazioni di configurazione con riferimento ad altri file. Nella parte finale invece (che inizia con la riga: `twm &`), esso definisce lo Window Manager che parte di default all'avvio del server X. Se la sezione appare come segue:

```
# Start the window manager:
startkde
```

allora ciò significa che il comando `startx` farà partire KDE. Come abbiamo visto, tale settaggio può essere modificato manualmente o con `xwmconfig`.

5.2 Gli strumenti di configurazione

Per ottimizzare la resa dello schermo, dobbiamo impostare alcuni elementi del server grafico. Mentre Slackware 9.1, che adottava XFree ci offriva a tale proposito tre strumenti (`xf86config`, in modalità testuale; `xfree86setup`, in modalità pseudo-grafica; `xf86cfg` in modalità grafica), in Slackware 11.0, che adotta Xorg, si può lanciare `xorgconfig` in modalità testuale, `xorgsetup` in modalità pseudo-grafica e infine `xorgcfg` in modalità grafica. Scegliamo la prima possibilità, forse la meno decorativa, ma sicuramente la più lineare: apriamo dunque una console e digitiamo `xorgconfig`. Ci si apre così un menu molto intuitivo, che però richiede una conoscenza minimale del nostro hardware.

5.2.1 Usare xorgconfig

Molte delle domande che il programma pone sono assolutamente intelligibili, e le risposte assai variabili a seconda del nostro hardware.

Il mouse

Si inizia con la scelta del mouse: se abbiamo un mouse PS2 con rotellina, scegliamolo battendo il corrispondente numero di protocollo (nel caso specifico, il numero 4), non abilitiamo invece l'emulazione per i tre bottoni; la rotellina avrà bisogno di un piccolo, ulteriore intervento per funzionare. Accettiamo il dispositivo di default (`/dev/mouse`).

Ricordiamo in questa sede che il modulo `psmouse`, atto a supportare i mouse di tipo PS2, è disabilitato, in quanto blacklistato in `/etc/modprobe.d/blacklist`, e cioè non viene caricato per evitare, in certi sistemi, che il mouse non funzioni più se si commuta la console. Pertanto è necessario eliminare (magari commentandola) l'invocazione a quel modulo presente in `/etc/modprobe.d/blacklist`, oppure caricarlo con l'aggiunta di un particolare parametro in `/etc/rc.d/rc.modules`, nella forma:

```
/sbin/modprobe psmouse proto=imps
```

che lo carica con un solo protocollo di base, tale da produrre più difficilmente il problema appena menzionato. Qualora infine si ricompili il kernel, è possibile inserire in forma statica il supporto al mouse.

necessario verificare se il mouse funziona correttamente

La tastiera

Scegliamo quindi il tipo di tastiera, solitamente una generica a 105 tasti o non funzioneranno i caratteri delle parentesi uncinatae < e > (il tasto corrispondente dovrebbe essere il 4) e la lingua (l'italiano dovrebbe essere il numero 31). Tralasciamo di settare una variante per la tastiera (che ci consentirebbe di abilitare un layout alternativo). In questo caso infatti avremmo che la sezione pertinente di /etc/X11/xorg.conf non risulterebbe:

```
Section InputDevice
    Identifier      Keyboard1
    Driver          kbd
    .
    .
    .
    Option XkbRules  xorg
    Option XkbModel  pc105
    Option XkbLayout it
EndSection
```

bensì:

```
Section InputDevice
    Identifier      Keyboard1
    Driver          kbd
    .
    .
    .
    Option XkbRules  xorg
    Option XkbModel  pc105
    Option XkbLayout it,fr,de
    Option XkbVariant basic,basic,basic
    Option XkbOptions grp:alt_shift_toggle
EndSection
```

dove si comprende facilmente che sono stati attivati due altri layout (francese e tedesco) e che per commutare la tastiera dall'uno all'altro si utilizza la combinazione (detta «gruppo») **Alt+Maiusc**.

Le opzioni di XKB

Le ulteriori opzioni XKB sono utili a rimappare il funzionamento della tastiera nel server X o per scambiare le funzioni di tasti diversi (ad esempio abilitare l'uso di un diverso gruppo di tasti di controllo o scambiare i tasti di controllo) e per gestire i led luminosi relativamente ai tasti. Si tratta verosimilmente di modifiche da non apportare a una prima installazione di Slackware. Tuttavia ne diamo conto in forma sintetica. Premettiamo che è possibile ottenere un'immagine delle mappature col comando `xkbprint`. Nel nostro caso, per ottenerla con determinate caratteristiche, diamo il comando:

```
xkbprint -color -label symbols -ll 1 -pict all :0 tastiera.ps
```

Va tenuto presente che di consueto noi digitiamo le minuscole e premiamo **Maiusc (Shift)** per ottenere le maiuscole. Secondo lo standard ISO 9995, la selezione di un simbolo ha luogo in base a delle più complesse combinazioni di tasti che consentono di ottenere quattro livelli diversi. In pratica, su uno stesso tasto ci può essere una lettera minuscola o maiuscola e altre due lettere o comunque due simboli differenti, in base alla selezione del livello appropriato.

Il primo livello della tastiera è quello che si ottiene senza premere alcun tasto di commutazione, mentre il secondo livello della tastiera è quello che si ottiene impostando il tasto **Maiusc**. Il terzo e il quarto livello (ottenibili rispettivamente con **Alt Gr** e con **Alt Gr+Maiusc**) consentono l'inserimento di simboli ulteriori. Ad esempio, al primo livello della tastiera italiana il tasto a dà «a», al secondo livello (**Maiusc**) «A», al terzo livello (**Alt Gr**) il simbolo «æ» e infine al quarto livello (**Alt Gr+Maiusc**) il simbolo «Æ». È però possibile modificare l'attribuzione dei simboli ai tasti, attivando dei layout alternativi di tastiera e commutarli mediante la pressione di tasti appropriati, detti *modificatori*.

Se si vogliono usare queste opzioni, ci apparirà un lungo elenco di possibilità per la modifica del gruppo:

```
Group Shift/Lock behavior 1 Right Alt key switches group while pressed
 2 Left Alt key switches group while pressed
 3 Left Win-key switches group while pressed
 4 Right Win-key switches group while pressed
 5 Both Win-keys switch group while pressed
 6 Right Alt key changes group
 7 Left Alt key changes group
 8 Caps Lock key changes group
 9 Shift+CapsLock changes group
10 Both Shift keys together change group
11 Both Alt keys together change group
12 Both Ctrl keys together change group
13 Control+Shift changes group
14 Alt+Control changes group
15 Alt+Shift changes group
16 Menu key changes group
17 Left Win-key changes group
18 Right Win-key changes group
19 Left Shift key changes group
20 Right Shift key changes group
21 Left Ctrl key changes group
22 Right Ctrl key changes group
```

Si può scegliere una di queste opzioni (ad esempio, digitando **3** avremo che il tasto **Win** servirà a cambiare gruppo), oppure passare direttamente al menu successivo, premendo **Invio**. Ci apparirà un nuovo menu, che gestisce il terzo livello:

```
Third level choosers 1 Press Right Control to choose 3rd level
 2 Press Left Alt key to choose 3rd level
 3 Press Right Alt key to choose 3rd level
 4 Press any of Alt keys to choose 3rd level
 5 Press Menu key to choose 3rd level
 6 Press any of Win-keys to choose 3rd level
```



```
7 Press Left Win-key to choose 3rd level
8 Press Right Win-key to choose 3rd level
```

In questo caso, ipotizziamo di selezionare l'opzione (la numero 5) per cui è il tasto **Menu** a farci andare al terzo livello. Possiamo scegliere dei controlli ulteriori o passare alla voce successiva, sempre con **Invio**. Qui possiamo decidere se il cambio di gruppo dev'essere segnalato dall'accensione di uno dei led luminosi a scelta. E via dicendo, finché abbiamo esaurito la reimpostazione.

In conclusione, possiamo attribuire a un determinato tasto o a una combinazione di tasti il compito di farci passare da una tastiera a un'altra, oppure di farci passare da un livello a un altro.

Il monitor

Per settare il monitor, accettiamo la frequenza di refresh orizzontale fra quelle proposte o scegliamo **11** per impostare quella del nostro monitor; stessa operazione per la frequenza verticale (qui è l'opzione 5 a consentirci di impostare la frequenza). Diamo un nome a piacere che identifichi il monitor.

La scheda video

Scegliamo la scheda video dal database, quindi la sua memoria, infine attribuiamole un nome a piacere che la identifichi. Se i modi di risoluzione sono corretti, accettiamo. Specifichiamo infine la profondità di colore del monitor. Questi sono tutti dati di cui dovremmo già essere in possesso, e che sono comunque riportati nei manuali dello schermo e della scheda. Alla fine salviamo il file (di solo testo) contenente tutte le impostazioni, che è `/etc/X11/xorg.conf`.

5.2.2 Come modificare `xorg.conf`

Errori anche banali possono impedire al server X di funzionare. In questo caso è necessario rieditare il file `/etc/X11/xorg.conf` con un editor senza poter entrare in modalità grafica, cioè con editor quali Vi (per una rassegna delle funzioni base di Vi, vedi [Scheda n. 10 – L'editor Vi](#)).

Scheda n. 10 L'editor Vi

Gli editor di solo testo di GNU/Linux sono numerosissimi. Alcuni richiedono il supporto del server grafico, come Kwrite o Kate, altri invece funzionano da console, come Pico, Joe, Emacs, Vi, Vim (che ha anche una versione grafica che gira sotto X) e Nano (quest'ultimo non incluso in Slackware). I due editor più classici – intorno a cui non mancano le dispute per la palma di «miglior editor» – sono Vi (che ha peraltro numerosi cloni: quello adottato in Slackware è Elvis, come si evince dal fatto che `/usr/bin/vi` non è che un link simbolico per l'appunto a Elvis) ed Emacs.

Qui offriremo soltanto alcuni cenni base su Vi, che può ritenersi uno strumento efficacissimo per modificare i file di configurazione, che sono di solo testo, anche se il server X fosse impossibilitato a funzionare. D'altro canto, Vi è l'editor la cui presenza è immancabile in qualsiasi distribuzione GNU/Linux e Unix in generale; una volta che si sia familiarizzato almeno coi comandi principali, la composizione e la modifica di un testo risultano molto veloci, e anzi proprio la velocità è uno dei contrassegni specifici di questo editor. Delle moltissime funzioni di Vi annoveriamo la possibilità di creare comandi personalizzati, di utilizzare script, di adottare vari schemi di indentazione automatica relazione ai linguaggi usati.

La schermata di Vi

Vi si attiva lanciando da console il comando `vi`. Se seguito dal nome di un file esistente lo apre, se seguito dal nome di un file non ancora esistente, lo crea. Infine, quando si apre Vi senza specificare nulla, ci si trova davanti un file (o meglio buffer) vuoto. Vi non presenta alcun menu e, contrariamente a molti editor, funziona secondo due diverse modalità: la modalità di comando e la modalità di inserimento di testo. Esse sono riscontrabili in basso a destra nel buffer, indicate rispettivamente con **Command** e **Input**. I numeri che precedono queste sigle indicano la riga e il carattere del testo (ad esempio, 3*5 indica che il cursore è attivo sulla quinta lettera della terza riga).

I «mode» di Vi: command mode e insert mode

Lanciato Vi, si apre un buffer vuoto e ci si trova nel cosiddetto «command mode», cioè una modalità in cui Vi si aspetta di ricevere istruzioni su che cosa fare (salvare, uscire, inserire testo, ecc.). Se si digita `vi` seguito dal nome di un file, quest'ultimo viene creato se inesistente, aperto se già esistente. Si può avviare Vi anche da una determinata riga di testo con l'opzione `+numero_di_riga`. Ad esempio, il comando: `vi +20 /etc/lilo.conf` apre `/etc/lilo.conf` alla riga 20, posizionando lì il cursore. La versione di Vi presente in Slackware consente anche l'apertura di buffer multipli (cioè di più file contemporanei). Dall'interno del file aperto, in command mode, bisogna dare il comando: `:split nome_file`. Il buffer viene così diviso in due: sotto sta il primo file aperto, sopra il secondo. Con un solo file aperto, `:split` divide in due lo schermo consentendo di accedere contemporaneamente a due parti del medesimo file.

In command mode si definiscono i comandi che vogliamo passare a Vi. Il primo è quello che determina lo spostamento all'«insert mode», cioè la modalità per scrivere nuovo testo. Va premuto il tasto **i** per scrivere dal punto di inserimento, oppure il tasto **a** per scrivere immediatamente dopo il punto di inserimento. Una funzione particolare dell'insert mode è il **replacement mode**, che non aggiunge, ma sostituisce il testo sovrascrivendo quello eventualmente presente. Per passare dal command mode all'insert mode e quindi da questo al replacement mode basta premere ogni volta **Ins**. Per tornare da un qualche modo definito al command mode è sufficiente premere **Esc**. Qualcuno è giunto addirittura a distinguere il command mode, che si ha quando il cursore è posizionato sul testo ma la tastiera si può utilizzare solo per richiedere l'esecuzione di comandi e non per inserire testo, dal directive mode, che si ha quando, mediante `:`, il cursore è posizionato nella «linea direttiva», che è poi l'ultima del video, pronto a scrivere i comandi (si tratterebbe insomma dell'analogo del minibuffer di Emacs).

I comandi di Vi

Vi è un programma molto potente, tutti i suoi comandi (dei quali qui ricorderemo solo i più immediati e utili) ammontano a qualche centinaio.

Fra i primi comandi da apprendere, ci sono quelli per muoversi all'interno del file. In alcune versioni di Vi, fra cui quella presente in Slackware, ci si può spostare coi tasti freccia e con i tasti **Inizio** e **Fine**; in altre versioni bisogna fare ricorso ai tradizionali tasti **h**, **j**, **k** e **l**. Molti comandi del command mode iniziano col doppio punto (`:`), in quanto tale simbolo indica al programma che quanto segue è da interpretarsi come un comando. Alcuni comandi possono essere combinati. Ad esempio, `10k` muove in altro dieci caratteri e `4dd` cancella quattro righe, la riga dov'è posizionato il cursore più le tre successive.

Alcune opzioni si impostano con comandi univoci, altre richiedono dei parametri numerici. Ad esempio, l'opzione `number` (che visualizza all'inizio di ogni riga il suo numero e che si imposta col comando `:set number`) non ha bisogno di parametri, mentre l'opzione `textwidth` (che si imposta col comando `:set textwidth=nn`) richiede di precisare il numero delle righe.

Per eseguire un comando di shell dall'interno di `vi` si usa il comando `!:comando [opzioni]`, ad esempio `!:ls -l`. Per uscire dal comando si batte **Invio**.

La scrittura con Vi

In insert mode, la scrittura si effettua normalmente da tastiera. È importante ricordare che Vi considera un paragrafo un testo delimitato da una linea vuota, e non da un semplice a capo. Di conseguenza, per evitare uno scorrimento lento del testo quando bisogna spostarsi, è buona norma impostare una fine

di riga fissa fra i settanta e gli ottanta caratteri. In questo caso l'a capo non sarà inteso come un paragrafo nuovo. Per ottenere questo risultato si ricorre a `:set textwidth=nn`, dove *nn* è il numero massimo di righe consentito; ad esempio, ponendo `:set textwidth=80` il testo andrà a capo automaticamente ogni 80 caratteri. Ricordiamo che `textwidth` presuppone che sia già attiva l'opzione `autoindent` (da porre col comando `:set autoindent`).

Poiché Vi non è un programma di videoscrittura come Writer di OpenOffice, non effettua alcuna rielaborazione delle modifiche alle righe e non ricalcola le interruzioni di linea. Pertanto, anche se abbiamo posto il margine a ottanta caratteri, quando cancelleremo delle parole da una riga, una parte della stessa resterà vuota. Per ragioni estetiche, si può riformattare il paragrafo al fine di ripristinare l'omogeneità del testo mediante l'operatore `gg`. Dopo aver selezionato un'area da riformattare in visual mode, si lancia l'operatore stesso. I parametri che si possono passare a questo operatore sono gli stessi che definiscono le porzioni di testo in Vi. Ad esempio, dando il comando `v4jgp` all'interno di una sezione di cinque righe col cursore posto sulla prima, si impartiscono in sequenza questi ordini: entrare in visual mode (`v`), selezionare dalla posizione del cursore alla fine della sezione (`4j`), lanciare il nostro operatore. Ma contare le righe spesso è assai scomodo. Possiamo allora riscrivere il comando precedente come: `j q}`, dal momento che `}` indica per l'appunto la fine del paragrafo. Per riformattare un paragrafo intero (posto che esso sia separato dagli altri da linee vuote) si darà il comando: `gqap`. Infine, per formattare un testo intero (sempre posto che tutti i paragrafi siano separati da linee vuote, altrimenti essi verranno uniti), si darà il comando: `ggqG` (`gg` per spostarsi all'inizio del testo e `G` per definire la fine del testo).

Vi apporta le modifiche su una copia del file in scrittura finché non si salva il testo. Ma è possibile anche salvare le modifiche a un file in sola lettura forzando il salvataggio o uscire da un file modificato senza salvare: basta aggiungere alla fine del comando un punto esclamativo, nella forma `:q!`, `:w!` oppure `wq!`).

Per selezionare ampie porzioni di testo, è necessario entrare nel cosiddetto visual mode, ossia andare in command mode e premere **v**: il movimento delle frecce della tastiera evidenziano il testo selezionato. **V** seleziona righe intere, **Ctrl+v** seleziona aree rettangolari, utile nelle tabelle (si tratta della cosiddetta «modalità visuale blocco»).

Apparentemente, Vi non possiede una funzione taglia-incolla. Ma il testo tagliato viene in realtà memorizzato in un buffer e può essere incollato mediante **p** (che lo inserisce prima del cursore) e **P** (che lo inserisce dopo il cursore). Per copiare, si ricorre al comando **y**: nella sua forma più semplice esso copia, memorizzandolo, il carattere su cui è posto il cursore. Quindi ci si sposta nel punto di inserimento, e si adottano sempre **p** o **P**. Per copiare aree più estese di testo, si integra il comando **y**; così, **yw** seleziona il testo dal cursore a fine parola (cioè, per Vi, fino al primo spazio che trova) incluso anche lo spazio, e **ye** una parola senza lo spazio finale; **y\$** copierà sino a fine riga, e via dicendo. Per copiare ampie aree di testo, ci si vale della «modalità visuale blocco» e quindi si inserisce il testo ancora una volta con **p** o **P**. Per spostare tali aree, allo stesso modo le si definisce con la «modalità visuale blocco», le si cancella con **d** e quindi le si inserisce nella nuova posizione con **p** o **P**.

È possibile creare degli alias, ossia delle abbreviazioni atte a inserire velocemente del testo più lungo mediante il comando `ab`: con la sintassi: `:ab [alias] [testo]`. Gli alias creati in una sessione di Vi si rimuovono col comando: `:unab [alias]`.

Ricerca e sostituzione

La ricerca di stringhe è in Vi «case sensitive», per cui di default viene fatta differenza fra le maiuscole e le minuscole. Si può sospendere questa funzione impostando il comando `:set ignorecase`, ripristinandolo successivamente con `:set noignorecase`. In questo caso, però, vengono rinvenute tutte le stringhe, indipendentemente dalla loro collocazione. Cercare la stringa `pera` porterà all'individuazione di «pera», ma anche di «imperatore», «operazione», «salperanno» ecc. Per cercare soltanto le parole, e non la stringa, si deve racchiudere la parola fra specifici operatori: si avrà perciò il comando: `/\, che fornirà solo le occorrenze della parola «pera». Nelle ricerche, il carattere punto (.) vale come il punto interrogativo nella shell, e dunque come un singolo carattere qualsiasi.`

La sostituzione può essere manuale o automatica. Nel primo caso ci si vale dei comandi **r** e **R**. Il comando **r** dev'essere lanciato col cursore sulla lettera da sostituire, seguito dalla nuova lettera. Il comando **R** consente di sostituire un numero a piacere di caratteri, per poi tornare alla modalità di comando digitando **Esc**. Nel secondo caso, il comando di base adotta la seguente sintassi: `:s/stringa_da_sostituire/stringa_sostitutiva`. Tale comando si limita a sostituire la prima occorrenza trovata di `stringa_da_sostituire` con `stringa_sostitutiva`. Per sostituire tutte le occorrenze in una riga il comando è: `:s/stringa_da_sostituire/stringa_sostitutiva/g`, dove `g` è l'opzione che sta per «global».

Altri operatori consentono di effettuare altri tipi di sostituzione, di cui offriamo qualche esempio. Il comando `:s/stringa_da_sostituire//` sostituisce la stringa con una stringa vuota, cioè, in effetti, la cancella. I comandi `:s/^/stringa` e `:s/$/stringa` aggiungono la stringa rispettivamente all'inizio e alla fine della riga. Il comando `:s/stringa1/&stringa2` concatena le due stringhe trasformandole in una stringa unica.

Infine, è possibile specificare l'area in cui effettuare la sostituzione. Il comando: `:n1,n2s/stringa_da_sostituire/textit stringa_sostitutiva` sostituisce la prima occorrenza di `stringa_da_sostituire` nell'intervallo di righe specificato in `:n1,n2`, mentre la sostituisce in tutte le occorrenze se si aggiunge, ancora una volta, l'opzione `g`. E poiché, con questa sintassi, `1` indica la prima riga e `$` l'ultima, il comando `:1,$s/stringa_da_sostituire/stringa_sostitutiva` sostituirà tutte le occorrenze di `stringa_da_sostituire` in tutto il testo.

Editing multiplo

Aperto un file, è possibile aprirne un secondo e altri ancora col comando `:e nome_file`, quindi passare dall'uno all'altro con `:e# [Invio]`. I file a disposizione dell'editing si vedono col comando `:args`.

Quando si hanno più file aperti, può essere più comodo vederli uno alla volta a schermo pieno e passare dall'uno all'altro, oppure dividere lo schermo per una visualizzazione simultanea. In questo caso, ci serviamo di `:split`. Senza argomenti, questo comando apre una nuova finestra sul file corrente dividendo a metà lo schermo. Se si aggiunge come argomento il nome di un altro file, i due file vengono aperti su una metà dello schermo ciascuno. Se si ha bisogno di una dimensione particolare, si ricorre al comando `:n split`, dove `n` è il numero delle righe del file che saranno visualizzate. Per aprire una nuova finestra il comando è invece `new`. Le dimensioni delle finestre possono essere aumentate o diminuite rispettivamente con `Ctrl-W + e` con `Ctrl-W -`. Per passare da una finestra all'altra si usa `Ctrl-w v`, per chiudere la finestra corrente `:close` e per chiudere tutte le finestre tranne quella corrente `:only`. Vi non supporta la divisione verticale delle finestre, attuabile invece in Vim. Per salvare le singole finestre, i comandi sono quelli tradizionali, mentre per gestirle tutte insieme c'è una serie di comandi atti a chiudere, salvare e uscire.

Controllo e personalizzazione

Per vedere le impostazioni correnti di Vi si digita `:set all`, mentre per controllare l'impostazione di un'opzione specifica `:set nome_opzione`. Invece `:set` elenca soltanto le opzioni modificate rispetto al default.

Inoltre, si può modificare stabilmente la configurazione di base scrivendo regole nuove nel file `.exrc`, contenuto nella home dell'utente (file di solo testo che in Slackware va creato dall'utente stesso, perché di default non è presente). Ad esempio, per far sì che Vi crei automaticamente un backup dei file modificati, che aggiungeranno all'estensione la sigla `.bak` (secondo l'impostazione di `/usr/share/elvis-2.2_0/elvis.bwk`), in `~/ .exrc` va aggiunta la riga: `set backup`. Aggiungiamo incidentalmente che, se può disturbare avere le directory con disseminati svariati file di backup, è possibile indirizzarli tutti a una apposita directory: in questo caso la riga risulterà: `set backup=directory_dei_backup` – ad esempio: `set backup=./backup`.

Imparare a usare Vi(m)

Per apprendere Vi, si può comunque usare il tutorial di Vim, disponibile anche in lingua italiana, per

sperimentarne in concreto il funzionamento; esso si attiva da shell col comando: `vimtutor [it]`. Se è stata effettuata l'italianizzazione di Slackware, il tutorial dovrebbe già partire nella lingua di sistema.

Vi presenta innumerevoli funzioni avanzate di cui qui non abbiamo tenuto conto, e molte altre aggiuntive sono presenti solo in Vim (ViIMproved), una versione migliorata e potenziata di Vi, anch'essa presente in Slackware. Per lavorare estesamente sul testo, ormai è consigliabile adottare Vim, impostando anzi in aggiunta la non compatibilità con Vi, per fruire di tutte le potenzialità di questo editor. Slackware 11.0 installa già una versione di `.vimrc` dotata di elementi di personalizzazione, e cioè `/usr/share/vim/vimrc`, basato su `/usr/doc/vim-7.0.109.doc/gvimrc_example.vim`. Esso costituisce un buon punto di partenza per creare un efficace file di configurazione ulteriormente personalizzato e va copiato nella home dell'utente come `.vimrc`.

Comandi principali di Vi

a) Operazioni sui comandi

Entra in insert mode (dal punto in cui è il cursore) = **i**

Entra in insert mode (dal punto successivo a quello in cui è il cursore) = **a**

Entra in insert mode (da inizio riga) = **I**

Entra in insert mode (da fine riga) = **A**

Ripete l'ultimo comando = **.**

Annulla l'ultima operazione = **u**

Ripristina l'ultima operazione cancellata = **Ctrl+t**

b) Operazioni sui file

Apri un file = **:e nome_file**

Salva con nome il file aperto = **:w nome_file**

Esce da Vi senza salvare il file corrente (se non modificato) = **:q**

Esce da Vi senza salvare il file corrente (se modificato) = **:q!**

Esce da Vi salvando il file = **:wq** oppure **ZZ**

Mostra informazioni sul file = **Ctrl+G**

Mostra il numero delle linee del file = **:=**

c) Spostamento del cursore

Muove a sinistra di un carattere = **h**

Muove in basso di un carattere (passare alla riga inferiore) = **j**

Muove in alto di un carattere (passare alla riga superiore) = **k**

Muove a destra di un carattere = **l**

Si sposta all'inizio della parola successiva = **:w**

Si sposta alla fine della parola successiva = **:e**

Si sposta all'inizio della parola precedente = **:b**

Si sposta a inizio riga = **^**

Si sposta a fine riga = **\$**

Si sposta a una riga determinata = **:numero_riga** oppure **numero_rigaG**

Si sposta a inizio paragrafo = **:{**

Si sposta a fine paragrafo = **:}**

Si sposta a inizio file = **:1**

Si sposta a fine file = **G**

d) Gestione del testo

Copia una riga = **yy** oppure **Y**

Copia il testo selezionato (solo in visual mode) = **y**

Inserisce la riga copiata dopo il cursore = **p**

Inserisce la riga copiata prima del cursore = **P**

Sostituisce una lettera (dov'è il cursore) = **rlettera**

Sostituisce più lettere (da dov'è il cursore) = **Rlettere**

Cancella una lettera dopo il cursore = **x**
 Cancella una lettera prima del cursore = **X**
 Cancella *n* lettere = **nx**
 Cancella la parola dov'è il cursore = **dw**
 Cancella una parte di riga (dal punto di inserimento del cursore) = **D**
 Cancella una riga (dov'è il cursore) = **dd**
 Cancella *n* righe = **ndd**
 Unisce una riga con la successiva = **J**
 Inserisce una riga vuota sopra quella dov'è il cursore = **o**
 Inserisce una riga vuota sotto quella dov'è il cursore = **O**
 Cambia le maiuscole in minuscole e viceversa: ~ (che si ottiene con **Crtl-Alt-^**).
 Inserisce il testo di un altro file = **:rnome_file_da_inserire**
 Interisce l'output di un comando da shell = **:r!comando_da_shell**
e) Ricerche sul testo
 Cerca una stringa di testo successiva = / **testo**
 Continua la ricerca in basso = **n** oppure / **[Invio]**
 Cerca una stringa di testo precedente = ' **testo**
 Continua la ricerca in alto = **N** oppure ' **[Invio]**
 Sostituisce una parola = **:s/parola_vecchia/parola_nuova [Invio]**
 Inserisce un segnalibro in una riga = **:m segnalibro**
 Va a un segnalibro = **:'segnalibro**
f) Gestione delle finestre
 Apre una nuova finestra sul file corrente = **:split**
 Apre una seconda finestra col file nuovo dividendo orizzontalmente lo schermo = **:split nome_file**
 Apre una finestra alta *n* righe = **:nsplit nome_file**
 Apre una seconda finestra in un file nuovo (vuoto) = **:new**
 Aumenta le dimensioni della finestra corrente = **Ctrl-W +**
 Riduce le dimensioni della finestra corrente = **Ctrl-W -**
 Passa all'altra finestra = **Crtl-w v**
 Chiude la finestra corrente = **:close**
 Chiude tutte le finestre tranne quella corrente = **:only**
 Chiude tutte le finestre = **:qall**
 Salva tutte le finestre = **:wall**
 Salva tutte le finestre ed esce = **:wqall**
 Esce ignorando i cambiamenti = **:qall!**
g) Ottenere aiuto
 Apre la guida in linea (in formato info) = **help**
 Visualizza la lista dei comandi di Vi = **:help index**
 Visualizza la guida su un determinato comando, opzione o codice di errore = **:help stringa**
 Lancia il tutorial = **vimtutor**

5.3 Ricorrere a Xvidtune

Xvidtune è un'applicazione per l'ambiente X Window che permette di configurare le frequenze video del monitor e di operare sulla centratura (le cosiddette «modeline»). In altre parole, consente di effettuare la regolazione fine di tutti i timing controllabili sull'uscita video, di spostare e/o ridimensionare l'area visibile dello schermo e inoltre fa vedere i refresh rate, suggerendo eventuali modifiche da appor-

tare poi manualmente a `/etc/X11/xorg.conf` per forzare determinati timing a determinate risoluzioni in determinati video mode.

Si lancia da console col comando `xvidtune`. L'utilità consente di regolare la giusta geometria dello schermo (le frequenze orizzontale e verticale) e di vedere subito il risultato (ricorrendo al pulsante **Apply**). A questo punto il pulsante **Show** mostra sulla console una serie di numeri come:

```
1280x1024 135.00 1280 1296 1440 1688 1024 1025 1028 1066 +hsync +vsync
```

che indicano le frequenze: tale stringa (che si può visualizzare direttamente senza anteprima col comando: `xvidtune -show`) va copiata nella sezione `Monitor` di `/etc/X11/xorg.conf`, premettendole la stringa `Modeline`, ossia:

```
Modeline 1280x1024 135.00 1280 1296 1440 1688 1024 1025 1028 1066 +hsync +vsync
```

5.4 Abilitare la rotellina del mouse

Una volta salvato `/etc/X11/xorg.conf`, dobbiamo introdurre una modifica a mano per abilitare la rotellina del mouse. Apriamo il file con un editor di testo (i più semplici, in questo momento, sono per noi Kate o Kwrite). Andiamo alla sezione "Section Input Device", e nella sottosezione # Identifier and driver, relativamente alla voce Identifier "Mouse1", se possediamo un mouse PS2 modifichiamo la riga:

```
Option 'Protocol' 'PS/2'
```

in

```
Option 'Protocol' 'IMPS/2'
```

e quindi aggiungiamo sotto le righe:

```
Option 'ZAxisMapping' '4 5'  
Option 'Emulate3Buttons' 'no'
```

Altri tipi di mouse potrebbero richiedere modifiche diverse.

5.5 Abilitare l'accelerazione grafica nativa

Se è abilitata l'accelerazione grafica nel kernel dobbiamo decommentare (cioè togliere il cancelletto #), sempre in `/etc/X11/xorg.conf` nella sezione Section "Module", le righe:

```
Load 'glx'
```

e

```
Load 'dri'
```

e l'intera sezione DRI, ossia le righe:

```
Section 'DRI'
Mode 0666
End section
```

Salviamo e usciamo. Perché la modifica apportata abbia effetto, è necessario uscire da X e rientrarvi. Nel caso si ricompili il kernel, è necessario porre la dovuta attenzione a selezionare il chipset corretto per la propria scheda madre. Per individuarla ci si può valere del comando: `lspci -vv | grep PCI`. Ulteriori dettagli si possono trarre dai comandi: `dmesg | grep agpgart` oppure `dmesg | grep agp`. Ciò dovrebbe fornire sufficienti informazioni per compilare il modulo corretto.

5.6 Installare i driver per l'accelerazione grafica (schede ATI)

5.6.1 Scaricare i driver

Risultati migliori nell'accelerazione dovrebbero ottenersi installando i driver specifici per la propria scheda video. Prenderemo qui in esame il caso delle schede ATI Radeon. Va tuttavia posta una certa attenzione, perché talvolta il rilascio dei driver ATI è in ritardo rispetto alle versioni di Xorg e dobbiamo assicurarci della compatibilità fra la versione dei driver e quella di Xorg. Poiché tali driver producono dei problemi con Xorg su alcuni sistemi che usano DRI, nella directory `/extra` Slackware fornisce dei driver patchati che potrebbero risolvere. Comunque, allo stato attuale, ATI rilascia driver per varie versioni di Xfree e per Xorg.

Il driver proprietario si scarica all'indirizzo: <http://www.ati.com/support/driver.html>, mentre un driver open source è disponibile all'indirizzo: <http://www.schneider-digital.de>, a seconda del modello della scheda. Ne esistono due versioni: il semplice driver, nella forma di un eseguibile, e un installer che automatizza diverse operazioni.

Vediamo dapprima la procedura per installare il driver originale ATI su Xorg, poi quella per operare mediante l'installer. In ogni caso, nel kernel vanno disabilitate entrambe le voci che gestiscono l'accelerazione nativa, e cioè AGP e DRM, o quanto meno devono essere selezionate come moduli e non in forma statica. Con le ultime versioni dei driver rilasciate da ATI non è più necessario che nel sistema siano presenti i sorgenti del kernel, ma soltanto gli headers. Inoltre, dovranno essere presenti nella posizione originale i file `System.map`, `.config` e `bzImage` (i primi due stanno in `/usr/src/linux` e l'ultimo in `/usr/src/linux/arch/i386/boot`), che poi vanno a collocarsi, rinominati, in `/boot`.

5.6.2 Installazione manuale del driver

Il driver ATI è in formato `.rpm`, che Slackware è in grado di gestire, ma che, se possibile, è meglio evitare di installare direttamente perché il database dei pacchetti ne tenga conto. Trasformiamolo in `.tgz` col comando: `rpm2tgz fglrx_6_8_0-8.29.6-1.i386.rpm` (si tratta del più recente driver attualmente disponibile), quindi installiamolo col comando: `installpkg fglrx_6_8_0-8.29.6-1.i386.tgz`; il pacchetto dovrebbe risultare installato in `/lib/modules/fglrx`.

A questo punto passiamo alla compilazione e installazione, che dev'essere effettuata da console, senza che il server X sia attivo. Dovendo spostarci nella directory `/lib/modules/fglrx/build_mod`, lanciamo in sequenza i seguenti comandi:

- 1) `cd /lib/modules/fglrx/build_mod` (per spostarci nella directory dove compilare il driver)
- 2) `sh make.sh` (e lavorerà l'ATI Module generator per compilare il driver)
- 3) `cd ..` (per passare alla directory superiore)
- 4) `sh make_install.sh` (per installare il driver).

Se tutto è andato a buon fine, possiamo provare a caricare il modulo col comando: `modprobe fglrx` e col comando `lsmod` dovremmo vederlo caricato. Dobbiamo quindi aggiungere in `/etc/fstab` la seguente riga:

```
tmpfs /dev/shm tmpfs defaults 0 0
```

Per caricare il driver all'avvio, dobbiamo lanciare l'utilità di configurazione `aticonfig` (che ha sostituito il vecchio `fglrxconfig`), oppure apportare a mano alcune modifiche al file `/etc/X11/xorg.conf`, che preesiste. Nel primo caso, si lancia il comando `aticonfig --initial --input=/etc/X11/xorg.conf`, presupponendo che `/etc/X11/xorg.conf` esista già (laddove `fglrxconfig` lo ricreava *ex novo*). In questo modo risulteranno abilitate le voci relative ai moduli GLX e DRI, mentre bisognerà decommentare manualmente le righe relative ai permessi di DRI, per conferire a tutti gli utenti il permesso 666:

```
# Section "DRI"
#     Mode 0666
# EndSection
```

Se invece vogliamo mantenere il vecchio `/etc/X11/xorg.conf`, apportiamo le seguenti modifiche:
a) alla Section `Modules` devono risultare decommentate le righe:

```
#This load the GLX modules
load 'glx'
#This load the DRI modules
load 'DRI'
```

b) alla sezione `Graphic device section`, Section `"Device"` è necessario sostituire il driver corrente (ad esempio `radeon`) con `fglrx`.
c) la sezione sopra riportata Section `"DRI"` dev'essere decommentata per intero.

5.6.3 Installazione automatica del driver

Se invece di preferisce fare ricorso all'installer, esso andrà lanciato (a seconda di quale sia la versione) col comando `sh ati-driver-installer-8.29.6.run` e attendere la decompressione. Sceglieremo quindi la prima voce del menu, e cioè «Install driver 8.29.6 on Xorg 6.9.x». Alla questione «Please, select the class of installation» scegliamo «Recommended» e attendiamo che si concluda l'operazione. A questo punto possiamo lanciare o meno un browser testuale (Lynx) per leggere le informazioni. Se invece optiamo per l'installazione «Expert», forniremo il path per i driver, che sarà sempre e comunque `/lib/modules/fglrx` e sceglieremo i componenti da installare; sono essenziali i `Display and OpenGL drivers` e i `Kernel Modules`, mentre sono opzionali le `Release Notes` e il `Control Panel`.

Se si opta per la seconda voce del menu, ossia «Generate Distribution Specific Driver Package», che non crea correttamente il `.tgz` per Slackware (in questo caso giova valersi della procedura seguita al punto precedente), il driver verrà installato, ma non risulterà alcun pacchetto nel database. Dato l'OK per la configurazione e approvata la Licenza, si fornirà al programma il path per i driver, si salteranno le opzioni, e infine si sceglierà la voce `Ati Drivers` (in quanto le altre sono pacchetti specifici per altre distro GNU/Linux, che non includono i `.tgz`). Si sceglierà infine la versione di Xorg e quindi basterà attendere il completamento dell'operazione.

Verrà in ogni caso creata una voce di menu in `/opt/kde3/share/applnk`, che tuttavia risiede in una posizione non standard. Se dunque si vorrà avere un menu grafico di configurazione in KDE, tale voce andrà spostata manualmente in `/opt/kde/share/applnk`.

5.6.4 Verifica dell'installazione

A questo punto rilanciamo il server X con `startx` e verifichiamo se tutto funziona: il comando `fglrxinfo` deve restituirci un output del tipo:

```
display: :0.0 screen: 0
OpenGL vendor string: ATI Technologies Inc.
OpenGL renderer string: RADEON 9000 DDR Generic
OpenGL version string: 1.3 (X4.3.0-3.9.0)
```

Se la riga `OpenGL vendor string: ATI Technologies Inc.` riportasse la voce `driver MESA` anziché `ATI`, questo significherebbe che i driver corretti non sono stati caricati.

Il comando `glxinfo` deve darci alle prime righe un output del tipo:

```
name of display: :0.0
display: :0 screen: 0
direct rendering: Yes
```

Se invece leggessimo `direct rendering: No`, ancora una volta ciò significherebbe che qualcosa non ha funzionato.

Infine, il comando `glxgears` ci mostra i frames per secondo, con l'aiuto di una finestra grafica a 2D, mentre il comando `fgl_glxgears` ci mostra i frames per secondo in una finestra a 3D.

5.6.5 Errori eventuali

Se il server grafico non parte più e riceviamo un avviso del tipo:

```
(WW) fglrx: No matching Device section for instance (BusID PCI:1:0:0) found
(EE) No devices detected
Fatal server error:
no screens found
.
.
.
XI0: fatal error 104 (Connection reset by peer) on X server ':0:0'
after 0 request (0 known processed) with 0 event remainig
```

basta entrare da console con un editor in `/etc/X11/xorg.conf`, individuare la `Graphics device section` e, nella sottosezione "Misc Options", commentare la riga del tipo:

```
BusID 'PCI:1:0:1' # vendor=1002, device=496e
```

a condizione che nel sistema sia presente un solo schermo.

5.7 Installare i driver per l'accelerazione grafica (schede Nvidia)

5.7.1 Scaricare e installare il driver

I driver per le schede Nvidia sono di semplice installazione, e i risultati, quanto all'accelerazione grafica, buoni. Essi sono reperibili all'indirizzo: <http://www.nvidia.com/content/drivers/drivers.asp> e, nel sito di lingua italiana, all'indirizzo: <http://www.nvidia.it/page/drivers.html>.

Una volta scelto il driver opportuno per la propria scheda, un eseguibile (script) di estensione `.run`, è necessario effettuare tutte le operazioni qui descritte da root in modalità testuale (se il server X è attivo, l'installazione fallisce) e con una connessione a Internet possibilmente attiva, almeno per la prima volta:

- 1) dalla directory in cui è presente il driver, trattandosi di uno script eseguibile si lancia il comando di installazione nella forma: `sh ./nome_file.run`.
- 2) si accetta il contratto («Licence»).
- 3) se l'installazione viene effettuata per la prima volta, apparirà nella maggior parte dei casi un messaggio che avverte:

```
No precompiled kernel interface was found to match your kernel:
would you like the installer to attempt for download a kernel interface
for your kernel from the NVIDIA ftp site (ftp://download.nvidia.com)?
```

È opportuno rispondere «yes» se il computer è già connesso a Internet, e inizierà il rapido download.

- 4) Se, com'è nella maggioranza dei casi, il programma non reperisce un'interfaccia precompilata adatta al kernel, apparirà un successivo messaggio:

```
No matching precompiled kernel interface was found on the NVIDIA ftp site; this means
that the installer will need to compile a kernel interface for your kernel.
```

Anche in questo caso va dato l'OK.

- 5) Si deve quindi attendere la costruzione del nuovo modulo, avvertiti dal messaggio: «Building kernel module. Installing Nvidia accelerated driver». Quando il processo è compiuto, un nuovo messaggio avverte: «Installation of Nvidia Accelerated Driver for linux-x86 is now complete». La documentazione relativa è consultabile in `/usr/share/doc/NVIDIA_GLX_1.0/README.txt`.

- 6) Ricordiamo che i driver Nvidia, per poter essere compilati, richiedono che kernel e moduli del kernel siano compilati con lo stesso compilatore.

5.7.2 Configurare `xorg.conf`

A questo punto, non resta che configurare `/etc/X11/xorg.conf`. L'unica operazione essenziale è modificare il nome del driver da quello che era (`nvd`, se si è installato quello presente di default con `xorgconfig`) in `nvidia`, per cui in “Graphics device section”, alla voce “Section Device”, dovrà risultare, ad esempio:

```
# *****
# Graphics device section
# *****
.
.
.
Section "Device"
Identifier "Nvidia GF6600"
Driver      "nvidia"
.
.
.
EndSection
```

È inoltre possibile effettuare una configurazione più raffinata aggiungendo ulteriori parametri nella medesima sezione, sotto al nome del driver, con un editor qualsiasi di testo. I parametri applicabili sono innumerevoli e indicati nel sopra citato file README.txt, ma i tre più utili e immediati (da attivare con la stringa booleana true) sono:

```
Option "NoLogo" "true"
Option "AllowGLXWithComposite" "true"
Option "RenderAccel" "true"
```

Il primo disabilita lo splash screen con il logo Nvidia che altrimenti apparirebbe a ogni apertura del server X.

Il secondo abilita GLX quando è caricata l'estensione Composite X. Tuttavia, questo parametro può generare dei problemi con le applicazioni OpenGL, per cui di default GLX è disabilitato quando Composite è caricato.

Il terzo abilita l'accelerazione per le estensioni RENDER; esso tuttavia è sperimentale e va pertanto testato macchina per macchina.

Poiché si tratta di un driver esterno al kernel, la sua installazione dovrà essere effettuata dopo ogni ricompilazione del kernel.

Scheda n. 11

Il server X e il file /etc/X11/xorg.conf

Il sistema X Window fornisce l'interfaccia grafica della maggior parte delle distribuzioni GNU/Linux. X adotta un modello client/server (anche in locale), dove i client sono semplicemente dei programmi che si rivolgono al server per la gestione dei display. Il server X disegna finestre, pulsanti ecc. in risposta alle chiamate del client, che non dev'essere necessariamente proprio dell'hardware locale. Ciò significa che si può creare una rete con più computer che vedono questi client da macchine diverse, e inoltre che è possibile anche esportare un display, cioè visualizzare sulla propria macchina programmi in esecuzione su macchine remote (ciò, peraltro, non solo all'interno di sistemi GNU/Linux, ma, con apposite utilità, anche fra sistemi operativi diversi come MacOSx e Windows).

Attualmente Slackware adotta Xorg, una implementazione open source liberamente redistribuibile dello standard X11. Esso è un'infrastruttura desktop basata su X11. Il server X primario include numerose ottimizzazioni tecnologiche, fra cui il supporto per l'accelerazione dell'hardware, l'estensione

XRender per l'anti-aliasing dei font, un design basato su driver modulari, un supporto aggiornato per dispositivi video hardware e di input, e numerose altre caratteristiche.

Dal punto di vista dell'utente, il file più importante di questo pacchetto (o meglio serie di pacchetti) è `/etc/X11/xorg.conf`, che raccoglie tutta la configurazione del server X. È un file di solo testo, diviso in una serie abbastanza complessa di sezioni, generato dall'utilità `xorgconfig`, ma che, al limite, potrebbe anche essere redatto a mano per intero. In ogni sezione vengono specificati dei parametri o caricati dei moduli per il funzionamento di X. Ogni sezione si individua facilmente, perché inizia con la riga: `Section "<section-name>"` e termina con la riga: `EndSection`. All'interno di ciascuna sezione sono presenti diverse righe contenenti un nome opzione e almeno un valore d'opzione, a volte fra virgolette.

Sezione MODULE contiene l'indicazione dei moduli della directory `/etc/X11R6/lib/modules` da caricare per il codice di supporto del display, fra cui quelli per i font ed (eventualmente) quelli per l'accelerazione grafica.

Sezione FILES presenta dapprima il percorso di RGB, il database dei colori (indicati per nome nel file `/usr/X11R6/lib/X11/rgb.txt`); la sigla è l'acronimo di Red Green Blu (rosso verde blu), i tre colori fondamentali che definiscono tutti gli altri sul monitor. Quindi contiene i path dei moduli che consentono di impostare directory multiple per l'archiviazione dei moduli caricati dal server Xorg. Infine elenca le diverse directory dei font (la cui configurazione sta in `/etc/X11/fs/config`, file di solo testo che però non dev'essere mai modificato a mano) o il numero di porta del font server.

Sezione SERVER FLAGS contiene alcune opzioni particolari di X, in genere inattive (le righe sono infatti tutte commentate e tali devono restare per l'utente inesperto).

Un po' più in dettaglio, le opzioni sono:

NoTrapSignals utile per fare eventualmente il debug.

DontVTSwitch disabilita la possibilità di passare da una console virtuale all'altra riportando i tasti funzione al loro uso normale.

DontZap disabilita l'uso della combinazione di tasti **Ctrl-Alt+Backspace** atta a uscire forzatamente da X (e da usarsi solo in caso di emergenza).

Dont Zoom disabilita l'uso della combinazione di tasti **Ctrl-Alt++** e **Ctrl-Alt--** atta a cambiare al volo la risoluzione dello schermo (la prima aumentandola, la seconda diminuendola).

DisableVidModeExtension disabilita le estensioni VidMode (Video Mode), che attraverso il client `xvidtune` cambiano in X alcune impostazioni dello schermo (fra cui le frequenze orizzontali e verticali).

AllowNonLocalXvidtune consente di usare `xvidtune` anche in base alle direttive di un host diverso da quello locale.

DisableModInDev disabilita la possibilità di modificare dinamicamente, al volo, le impostazioni delle periferiche di input (mouse e tastiera).

AllowNonLocalModInDev consente di modificare le impostazioni di mouse e tastiera da remoto (mediante l'utilità `xset`).

Vi sono inoltre altre opzioni che possono essere aggiunte manualmente.

Sezione INPUT DEVICE contiene la configurazione delle periferiche di input, e usualmente non manca di almeno due sottosezioni, una relativa alla tastiera (Core keyboard's InputDevice section) e l'altra al mouse (Core Pointer's InputDevice section). Per ogni sottosezione vengono precisati il nome che identifica il dispositivo e il driver che utilizza, oltre a opzioni aggiuntive facoltative. La seconda parte di questa sottosezione contiene l'eventuale personalizzazione della tastiera, cioè l'attribuzione di funzioni particolari a determinati tasti. Infine è presente un'ultima sottosezione per dispositivi aggiuntivi.

Sottosezione CORE KEYBOARD'S INPUTDEVICE contiene la configurazione della tastiera. Non solo dunque le informazioni già presenti nella sezione precedente, ma anche quelle sul modello e il layout per la corretta attribuzione dei tasti. Inoltre specifica ritardo e frequenza della ripetizione della tastiera (ossia per quanto tempo dev'essere premuto un tasto perché per avviare la ripetizione del carattere e quanti caratteri verranno introdotti al secondo). Si possono aggiungere qui ulteriori opzioni di xbd per l'attribuzione di funzioni particolari ai tasti e per far accendere i led luminosi in combinazioni a piacere in circostanze definite.

Sottosezione CORE POINTER'S INPUTDEVICE contiene la configurazione del mouse. Qui è possibile abilitare la rotellina e l'emulazione dei tre tasti. Inoltre è questa la sezione da modificare se si possiede un mouse con più tasti. Di norma i mouse PS/2 sono visti come il dispositivo `/dev/psaux`. Altri mouse (per esempio gli USB) sono visti come `/dev/input` o `/dev/input/mice`. È comunque possibile controllare se un certo dispositivo rappresenta effettivamente il mouse controllando l'output del dispositivo stesso da console col comando: `cat /dev/input/mouse0` mentre si muove il mouse, e quindi premere **Ctrl+C** per terminare la sessione.

Sezione MONITOR contiene l'identificativo dello schermo specificato nella sezione `ServerLayout Section`, cioè il nome a esso attribuito a piacere in fase di configurazione, e le sue frequenze orizzontali e verticali.

Sezione GRAPHICS DEVICE vengono qui definiti i parametri relativi al chipset della scheda video. Dapprima è presente una generica configurazione VGA; quindi alcuni elementi di configurazione che però servono solo se nel sistema è installato più di un video, per identificarne il BUS, cioè la collocazione fisica all'interno del computer. Nella sottosezione `Section "Device"` si leggono i valori immessi a proposito della scheda video durante la precedente configurazione. La riga relativa alla memoria video è commentata, in quanto di norma essa è stabilita automaticamente da Xorg.

Sezione SCREEN definisce una o più risoluzioni (la prima è quella caricata di default) e quindi la rispettiva profondità del colore. La risoluzione – come abbiamo visto – può essere cambiata anche mentre il server X è attivo.

Sezione SERVERLAYOUT qui viene stabilita la configurazione di base di X: sono definiti i layout del display e dello schermo, vengono assegnati i nomi ai dispositivi. A meno di non avere configurazioni hardware particolari, generalmente saranno presenti qui due sole righe, relative a `Keyboard0` e `Mouse0`, che indicano la tastiera e il mouse configurati di default per il sistema (come indicano le opzioni `CoreKeyboard` e `CorePointer`). Consente anche configurazioni multiple, se il sistema annovera più schermi o più periferiche di input (ad esempio, due mouse, come talora accade).

Sezione DRI questa sezione (la cui denominazione è l'acronimo di Direct Rendering Interface) gestisce le autorizzazioni per l'utilizzo dell'accelerazione grafica. Infatti, se decommentata, estende a tutti gli utenti questa possibilità conferendo a DRI i permessi 666.

5.8 Esportare il server X

5.8.1 Server e client

Esportare il server X significa poter vedere sul display di un computer i programmi attivati in un altro. Ciò può essere utile se un computer non è dotato di schermo, ma è connesso a uno che, invece, ne è dotato; ha inoltre una sua funzione anche per la medesima macchina, come vedremo subito dopo. Ciò può introdurre falle nella sicurezza del sistema, perché tutta la rete potrebbe vedere che cosa sta accadendo nel sistema, ma qui presupponiamo di essere all'interno di una rete sicura (altrimenti si dovrebbe adottare `ssh`). In secondo luogo, per evitare equivoci di terminologia, va aggiunto che, dal punto di vista della rete, il server è il computer che esegue i programmi e li invia al display dell'altro, mentre il client è il computer sul display del quale si possono vedere i programmi che stanno girando. Dal punto di vista del server X, invece, il server è il computer che visualizza i programmi, e il client è quello che glieli invia (qui, visto che parliamo del server X, adotteremo questa seconda opzione, ipotizzando che il server si chiami **darkstar** e il client **brightstar**).

5.8.2 Impostare il server

Il server dev'essere avviato con X attivo. Quindi si apre un terminale e si imposta il computer in modo che gli altri computer, i client, lo possano usare come display. A tal fine si adotta il comando: `xhost`. Per consentire un accesso illimitato a tutti gli utenti dei client, il comando sarà: `xhost +`, che fornirà un output del tipo:

```
samiel@darkstar: $ xhost +
access control disabled, clients can connect from any host
```

Se invece si desidera consentire questa funzione solo ad alcuni utenti, essi andranno precisati dopo il comando, nella forma `xhost + nome_computer` oppure `xhost + nome_utente`. In questo caso potremo avere:

```
samiel@darkstar: $ xhost + brightstar
brightstar being added to access control list
```

dove è stato concesso l'accesso al solo computer chiamato **brightstar**, oppure

```
samiel@darkstar: $ xhost + brightstar.sabina
brightstar.sabina being added to access control list
```

dove è stato concesso l'accesso al solo utente **sabina** del computer chiamato **brightstar**. Naturalmente, dev'essere attivo il servizio di risoluzione dei nomi nel file `/etc/hosts`. Non serve modificare altro, perché Slackware, a differenza di altre distribuzioni, non carica il server X con l'opzione `-nolisten tcp` (che altrimenti andrebbe disabilitata). È possibile verificare a quali utenti è stato concesso l'accesso lanciando il comando `xhost` senza ulteriori parametri. Per revocare i permessi, si adotta lo stesso comando col simbolo meno (-) in luogo del simbolo più (+).

5.8.3 Impostare il client

Il client deve sapere dove visualizzare i programmi: non sul suo monitor, ma su quello del server. A tal fine si deve modificare la variabile `$DISPLAY`, che in genere non ha alcun valore predeterminato, fornendole il nome del computer remoto e il numero della sessione X da usare come display. Se si usa la shell `bash` (altre adottano una sintassi lievemente diversa), il comando sarà: `export DISPLAY=darkstar:0.0`.

5.8.4 Lanciare i programmi

Ora è possibile lanciare un qualsiasi programma grafico dal client, che sarà visualizzato sul server, a patto, naturalmente, che in esso un server X sia attivo. Per eseguire il programma e lasciare al contempo libero il terminale, il programma verrà messo in background, per cui assumerà la forma: *comando &*.

5.8.5 Applicazioni grafiche da root in sessione utente

Xhost ha una qualche utilità anche all'interno dello stesso computer. Ipotizziamo di trovarci in una sessione utente col server X attivo e di voler lanciare un'applicazione grafica come root. Se ci limitiamo ad aprire una shell, diventare root e lanciare il programma, otterremo dei messaggi di errore. Ad esempio:

```
root@darkstar:~\# kate
kate: ERROR: KUniqueApplication: Can't determine DISPLAY. Aborting.
```

oppure:

```
root@darkstar:/usr/src/linux-2.6.10# make xconfig
scripts/kconfig/qconf arch/i386/Kconfig
qconf: cannot connect to X server
make[1]: *** [xconfig] Error 1
make: *** [xconfig] Error 2
```

Per risolvere il problema, essendo già loggati come utente, bisogna dapprima assegnare i permessi al client col comando: `xhost + root`. Quindi si passa a root nel modo consueto (con `su [-]`) e si assegna la variabile `DISPLAY` col comando: `export DISPLAY=:0`. A questo punto è consentito lanciare l'applicazione che gira sotto X come root. Se si adotta il comando: `xhost + localhost`, allora qualsiasi utente potrà usare le applicazioni.

Scheda n. 12

L'editor GNU Emacs

Insieme a Vi(m), GNU Emacs è il più prestigioso editor dei sistemi Unix, anche se in realtà è ricchissimo di ulteriori funzioni che lo rendono assai più di un semplice editor. Con Emacs è possibile, fra l'altro, inviare e-mail, consultare newsgroup su Usenet, fare dei giochi e addirittura ricorrere a un improbabile psicanalista. Inoltre, Emacs contiene delle facilitazioni per usare determinati linguaggi di scrittura o di programmazione, da C e C++ a Python, da Perl a L^AT_EX, e naturalmente Lisp, in cui è scritta la maggior parte del suo codice. Poiché dal suo interno è possibile effettuare anche la compilazione e il debug di codice, essi si presenta anche come un ambiente di sviluppo integrato. In Slackware si può avviare la versione testuale come pure quella supportata dal server X. Una versione avanzata (non presente in Slackware), e cioè (non-GNU) Xemacs, consente anche di gestire la grafica.

Benché anche come editor Emacs presenti moltissime funzioni avanzate (fra cui il controllo di subprocessi, la formattazione automatica di codice sorgente e il suo debugging, la gestione di espressioni e commenti in numerosi linguaggi di programmazione, la personalizzazione del suo aspetto e dei suoi

comandi, l'estensibilità, e cioè la possibilità di creare comandi completamente nuovi), qui offriamo solo alcuni cenni base, volti alla manipolazione semplice di file di puro testo. Inoltre, dal momento che abbiamo considerato Vi come editor da console, ora descriveremo alcune caratteristiche della versione di Emacs che gira sotto X.

La schermata di Emacs

L'editor si richiama semplicemente digitando `emacs` al prompt, e, se seguito dal nome di un file esistente lo apre, se seguito dal nome di un file non ancora esistente, lo crea. Infine, quando invece si apre Emacs senza specificare nulla, ci si trova davanti un buffer vuoto detto «scratch», dov'è possibile scrivere testo che non sarà salvato automaticamente (tranne che poi si impartisca l'ordine contrario). Nella schermata, che prende il nome di «riquadro» (frame), troviamo a sinistra la barra di scorrimento. In alto è posta la barra dei menu, attivabile col mouse da X, con tasti funzione da console. I file o comunque il testo aperto viene conservato in un'area chiamata «buffer»; il punto in cui, all'interno del buffer, è collocato il cursore del mouse viene chiamato «punto». Un buffer non coincide propriamente con un file: il testo elaborato nel buffer è una copia di quello originale e non lo modifica finché non viene salvato; inoltre vi sono buffer che vengono creati come risultato di determinati comandi. È possibile aprire più file in differenti buffer: avremo così molteplici riquadri nella medesima sessione di Emacs.

Nella parte inferiore dello schermo è presente una barra chiamata «riga di modalità», che fornisce informazioni come il nome del buffer corrente, la percentuale visibile sullo schermo, la riga in cui si trova il punto, l'eventuale file in cui viene salvato il buffer. Se si tratta di buffer a lettura e scrittura, troveremo i simboli `**` se esso è stato modificato, `--` se non lo è stato; se si tratta di buffer a sola lettura, troveremo i simboli `%*` se esso è stato modificato, `%%` se non lo è stato. La riga di modalità dichiara inoltre le modalità attive nel buffer: ad esempio l'insert mode, per cui il testo digitato viene inserito, l'overwrite mode, per cui il testo digitato sovrascrive il testo nel punto, nonché la modalità principale di scrittura.

Ancora più in basso troviamo la cosiddetta «area echo», dove Emacs comunica brevi messaggi di stato, come quelli di errore o altro. Quando si digita un comando che richiede che richiede un input, quest'ultimo viene richiesto e va digitato in quest'area. Questa riga è più nota col nome di «minibuffer», che in certo modo può essere descritta come la linea di comando di Emacs.

I comandi di Emacs

Molti comandi di Emacs (in realtà scorciatoie da tastiera per numerose funzioni che potrebbero essere invocate dal minibuffer) sono costituiti da una doppia combinazione che fa ampio ricorso al tasto **Ctrl**. Nella notazione specifica, **Ctrl** si scrive semplicemente **C**; ad esempio, poiché il comando per uscire dall'editor è **Ctrl+x Ctrl+c**, si avrà la formula **C-x C-c**. Le funzioni sono invece sempre precedute da **Alt+x**, che nella notazione risulta **M-x**, per il fatto che si riferisce al tasto **Meta**; in realtà è molto improbabile che le tastiere abbiano un simile tasto, e allora in suo luogo si utilizza **Alt**. Per eseguire una determinata funzione, si deve pertanto premere **M-x**, digitare nell'area echo il nome della funzione e infine premere **Invio**. Per uscire da una funzione o da un comando, si preme **C-g**, atto anche a interrompere e annullare l'input da tastiera se si commette un errore. Molti comandi accettano anche un argomento numerico, che in genere (ma vi sono importanti eccezioni, come **C-v** e **M-v**, che spostano il cursore avanti e indietro di n linee e non di n schermate) indica per quante volte vanno ripetuti. Tale argomento viene espresso dalla sequenza **C-u n**. Ad esempio, **C-u 5 c-f** sposta in avanti il cursore di cinque caratteri.

Molti comandi aprono un buffer nuovo, che a volte occupa una parte della finestra originaria, si può chiudere questa seconda finestra e tornare alla prima con **C-x 1**.

La gestione dei buffer di Emacs

Per aprire un file, si può ricorrere al menu oppure alla funzione `find-file`. Si digiterà, come scorciatoia da tastiera, **C-x C-f**, quindi si premerà **Invio**. Nell'area echo si scriverà infine il nome del file, considerando che il tasto **Tab** offre una scorciatoia per il completamento dei nomi, come in console.

Emacs può aprire contemporaneamente più buffer. Ad esempio, il comando: `emacs testo1.txt /etc/fstab /documenti/documento1` apre tutti e tre i file in questione. Per passare da un buffer al-

l'altro si usa **C-x b**: si passa all'ultimo file visitato (la visualizzazione di un buffer in Emacs è chiamata «visita») se non si precisa alcun nome, mentre per andare a un determinato buffer bisogna specificarne il nome.

Emacs consente di vedere contemporaneamente lo stesso buffer o più buffer in finestre diverse. Per suddividere il riquadro corrente in due finestre poste orizzontalmente si usa il comando: **C-x 2**; **C-x 3** divide ulteriormente la finestra attiva in due finestre accostate verticalmente. Le finestre possono essere ridimensionate col mouse. Per passare da una finestra all'altra o si clicca dentro col mouse, oppure si dà: **C-x o**. Il comando **C-x 1** chiude le finestre ripristinando quella principale.

Un buffer si chiude senza salvarlo (cioè eliminandolo, «killandolo») col comando **C-x k**. Se il contenuto proviene da un file, Emacs chiederà se lo si vuole salvare, se invece proviene da un buffer scratch, questo verrà chiuso direttamente.

Per uscire dal programma, si digita **C-x C-c**, che chiede in aggiunta se si vogliono salvare i buffer aperti e non ancora salvati. Per mettere il programma in background, si digita **C-z**: Emacs viene ridotto a icona e può essere ripreso in un secondo tempo.

La scrittura si effettua normalmente da tastiera. Quando una riga supera l'ampiezza dello schermo, un simbolo ci avverte che non si tratta se non della prosecuzione della riga precedente: una barra inversa o *backslash* in modalità console, una freccia ricurva in modalità grafica.

Ricerca e sostituzione

La selezione di aree ampie e uniformi di testo (che Emacs chiama «regioni») si effettua definendo l'inizio dell'area con un «mark point» con **C-@** oppure con **C-barra spaziatrice** e collocando alla fine il punto (in alternativa, si può selezionare la regione col mouse). Per copiare la regione (che viene collocata nel cosiddetto «kill-ring», qualcosa di simile a ciò che in altri editor di chiama *clipboard*, o «appunti») si dà **M-w** e quindi, per incollarla in un nuovo luogo, **C-y**. Per spostare la regione, si adotta **C-w** invece di **M-w**. Va ricordato che Emacs fa differenza fra eliminazione (*killing*) e cancellazione (*deleting*): il testo eliminato viene conservato in un buffer nascosto e può essere ripristinato, laddove il testo cancellato viene definitivamente eliminato ed è irrecuperabile.

Emacs offre diversi metodi per la ricerca e la sostituzione del testo. Il più semplice è la ricerca di stringhe che viene effettuata in modo incrementale. Si entra nella modalità di ricerca con **C-s** e quindi si scrivono (nel minibuffer) le lettere; terminata la stringa, si dà **Invio**. Emacs inizia subito a mostrare le occorrenze trovate a partire dal punto. Più lettere si scrivono, più si restringe il campo, finché viene individuata la prima occorrenza della stringa indicata. La ricerca si prosegue dando ancora **C-s**. La stringa **C-r** imposta invece una ricerca inversa, dal punto all'indietro.

La sostituzione si effettua col comando **M-x replace-string**. Nel minibuffer viene richiesta la stringa da sostituire e poi la sostituzione. Battuto **Invio**, tutte le occorrenze vengono sostituite. Se invece si vuole effettuare una sostituzione interattiva, in modo cioè che Emacs chieda a ogni occorrenza se effettuare o meno la sostituzione, si userà il comando **M-x query-replace**. Tutte queste sostituzioni avranno luogo a partire dal punto.

Modalità primarie, secondarie e personalizzazione

Come abbiamo detto, Emacs può scrivere linguaggi differenti, dal solo testo a svariati linguaggi di programmazione, fino ai file di $\text{T}_{\text{E}}\text{X}$, fornendo diverse tecniche di evidenziazione a colori, una diversa indentazione, un aiuto sensibile al contesto e si interfaccia automaticamente col debugger opportuno. Ogni buffer viene aperto in una «modalità primaria» a seconda del tipo di linguaggio, che ha un comportamento leggermento diverso dalle modalità primarie per altri linguaggi. L'help sulla modalità primaria attualmente attivata si apre con **C-h m**. Vi sono anche delle modalità secondarie, che integrano quella primaria con alcune utilità. Ad esempio, quando si è in modalità primaria text-mode (attivabile con **M-x text mode [Invio]**), si può attivare la modalità secondaria Auto Fill (attivabile con **M-x auto fill mode [Invio]**), la quale interrompe la riga quando essa diventa troppo lunga. Di default, l'interruzione di riga è fissata a settanta caratteri, ma può essere modificata col comando: **C-u nn C-x f** (dove *nn* è il numero

delle righe) e ripristinata con C-x f). Quando invece si scrive un determinato codice, la modalità auto-fill indenta diversamente il testo.

Infine, Emacs può essere altamente personalizzato. Il modo stabile più conveniente è editare un file `~/emac`, di solito modificando il valore di una variabile. Ad esempio, inserire in `~/emac` la stringa:

```
(setq inhibit-startup-message t)
```

serve a far sì che Emacs si apra direttamente con un buffer scratch vuoto, senza mostrare la finestra di benvenuto. Se si vuole rendere sempre attiva la modalità auto-fill, ipotizzando di fissare l'a capo all'ottantesimo carattere, si aggiungeranno in `~/emac` le seguenti righe:

```
(add-hook 'text-mode-hook 'text-mode-hook-identify)
```

```
(add-hook 'text-mode-hook 'turn-on-auto-fill)
```

```
(setq-default fill-column 80)
```

e via dicendo. I cambiamenti apportati mediante il menu **Option – Customize Emacs** sono salvati proprio in `~/emac`.

Imparare a usare Emacs

Emacs contiene un ampio tutorial che consente di apprendere tutte le sue funzioni fondamentali. Esso si avvia col comando **C-h t**. È disponibile in numerose lingue, compreso l'italiano.

La pagina man relativa a Emacs è molto sintetica, mentre una guida estesissima e dettagliata è disponibile in formato info. C'è anche la possibilità di ricorrere ad apropos per informazioni sulle funzioni se non si ricorda esattamente la stringa completa.

Il principiante è messo al riparo dal commettere errori con comandi potenzialmente pericolosi, i quali, di default, sono disattivati. Invocando tali comandi, si riceve un messaggio che domanda se attivarli o meno. Premendo la barra spaziatrice essi verranno comunque attivati, premendo **n** essi invece rimarranno disattivati.

Comandi principali di Emacs

a) Spostamento del punto

Sposta il punto alla riga precedente = **C-p** (previous)

Sposta il punto alla riga successiva = **C-n** (next)

Sposta il punto indietro (a sinistra) di un carattere = **C-b** (backward)

Sposta il punto avanti (a destra) di un carattere = **C-f** (forward)

Sposta il punto indietro (a sinistra) di una parola = **M-b**

Sposta il punto avanti (a destra) di una parola = **M-f**

Sposta il punto a inizio riga = **C-a**

Sposta il punto a fine riga = **C-e**

Sposta il punto all'inizio del paragrafo = **M-a**

Sposta il punto alla fine del paragrafo = **M-e**

Sposta il punto avanti di una schermata = **C-v**

Sposta il punto indietro di una schermata = **M-v**

Centra il testo rispetto alla riga dov'è il punto: **C-l**

Sposta il punto all'inizio del testo = **M-<**

Sposta il punto alla fine del testo = **M->**

b) Gestione del testo

Elimina il carattere a sinistra del punto = **C-d**

Elimina il carattere a destra del punto = **Canc**

Elimina la parola posta prima del punto = **M-Del**

Elimina la parola posta dopo il punto = **M-d**

Elimina il testo dal punto alla fine della riga = **C-k**

Elimina il testo dal segno al punto = **M-k**

Cancella tutto il testo = **C-l**

Inverte due caratteri = **C-t**
 Inverte due parole = **M-t**
 Definisce il segno = **C-@**
 Taglia il testo = **C-w**
 Copia il testo = **M-w**
 Incolla il testo = **C-y** (yanking)
 Trasforma in lettere maiuscole dal cursore a fine parola = **M-u**
 Trasforma in lettere minuscole dal cursore a fine parola = **M-l**
 Trasforma in lettere maiuscole la regione = **C-x C-u**
 Trasforma in lettere minuscole la regione = **C-x C-l**
c) Operazioni sui comandi
 Annulla l'ultima azione = **C-x u** oppure **C-_**
 Ripete *n* volte il comando corrente = **C-u n**
d) Operazioni sui file
 Salva un buffer in un file = **C-x C-s**
 Salva un buffer in un file con nome nuovo = **C-x C-w**
 Salva i buffer contenenti testo modificato chiedendo ogni volta conferma: **C-x s**
 Salva tutti i buffer aperti ed esce da Emacs = **C-x C-c**
 Apre un file in un nuovo buffer = **C-x C-f**
 Elenca i buffer aperti = **C-x- C-b**
 Imposta un segnalibro = **C-x r m**
 Passa a un segnalibro = **C-x r b**
e) Ricerche nel testo
 Cerca avanti di una stringa = **C-s**
 Cerca indietro di una stringa = **C-r**
 Sostituisce una stringa = **M-x replace-string**
 Sostituisce interattivamente una stringa = **M-%**
 Cerca e rimpiazza usando regexp = **M-x replace-regexp**
f) Menu speciali
 Visualizza in un menu tutti i buffer aperti (solo in X) = **C-dx del mouse**
 Visualizza un menu di selezione dei font (solo in X) = **Maiusc-dx del mouse**
f) Ricorrere all'help
 Avvia il tutorial = **C-h t**
 Descrive la pressione dei tasti specificati = **C-h x**
 Elenca i comandi relativi alla funzione specificata = **C-h f**
 Elenca i comandi relativi alla variabile specificata = **C-h v**
 Apre le FAQ in un nuovo buffer = **C-h F**
 Avvia Info = **C-h i**

Capitolo 6

Ottimizzare il server sonoro

6.1 Configurare il server sonoro

La configurazione del server sonoro richiede naturalmente che si sappia innanzitutto di quale scheda audio è dotato il computer, e quindi quale sia il suo chip. Della prima informazione si dovrebbe essere a conoscenza in base alle specifiche del proprio computer. Peraltro, se la scheda è già stata configurata in automatico con `alsaconf`, tale informazione si può ottenere col comando: `cat /proc/asound/cards`, il cui output sarà ad esempio:

```
0 [Live ]: EMU10K1 - Sound Blaster Live!  
Sound Blaster Live! (rev.10, serial:0x80651102) at 0xd800, irq 9
```

La seconda informazione si ottiene invece analizzando l'output del comando: `lspci [-v]`, dove, fra l'altro, dovremmo leggere qualcosa come:

```
00:0a.0 Multimedia audio controller: Creative Labs SB Live! EMU10k1 (rev 0a)  
Subsystem: Creative Labs SBLive! 5.1 Digital Model SB0220  
Flags: bus master, medium devsel, latency 32, IRQ 9  
I/O ports at d800 [size=32]  
Capabilities: [dc] Power Management version 1
```

che ci consente di capire che il chip della scheda audio è EMU10k1.

In primo luogo dobbiamo configurare la scheda sonora per far caricare i suoi moduli (richiamati in `/etc/modules.conf` con il kernel 2.4, e in `/etc/modprobe.d/sound` con il kernel 2.6).

Scheda n. 13

La gestione dei moduli

I moduli del kernel consentono a determinate periferiche di funzionare, oppure abilitano particolari funzioni nel sistema; normalmente, si distinguono i driver dei dispositivi, quelli per i file system e i driver di rete. Di solito, il nome dei file oggetto ha estensione `.o`, e così è stato anche per i moduli del kernel fino alla versione 2.4. Con il kernel 2.6, anche per facilitarne l'identificazione, i moduli hanno assunto l'estensione `.ko` (abbreviazione di «kernel object»). Il file `.ko`, in realtà, è composto da varie sezioni ed è il risultato del collegamento di due file: il vero e proprio file oggetto contenente il codice, di

estensione `.o`, e un file che contiene informazioni aggiuntive riguardo all'ambiente di compilazione del modulo, cioè un file oggetto ELF il cui nome adotta l'estensione `.mod.o`.

Tutti i comandi per la gestione dei moduli sono contenuti nel pacchetto **module-init-tools**, che implementa sia le **modutils** per operare coi kernel 2.4, sia i **module-init-tools** propriamente detti, per operare coi kernel 2.6.

Il vantaggio di un kernel modulare è che le voci abilitate per l'appunto in questa forma vengono caricate in memoria solo quando ve ne sia effettiva necessità. I moduli vengono collocati nella directory `/lib/modules/versione_kernel` ed elencati alla radice di questa directory nel file `modules.dep`, che ne individua anche le dipendenze; esso viene generato dal comando `depmod`. Il kernel è in grado di caricare automaticamente i moduli che determinati programmi invocano, qualora vi siano state abilitate, nella sezione `Loadable module support`, le voci: `Enable loadable module support` e `Automatic kernel module loading`.

In alcuni casi però è necessario intervenire manualmente per caricare o rimuovere un determinato modulo. Tutti i comandi relativi alla gestione dei moduli sono contenuti nel pacchetto **module-init-tools** (che integra sia le **modutils**, ossia gli strumenti di gestione per il kernel 2.4, sia i **module-init-tools**, ossia le utilità di gestione per il kernel 2.6).

Il comando: `insmod /percorso/nome_modulo` è il modo più semplice per inserire un modulo. Ad esempio, si ha: `insmod /lib/modules/2.6.18/kernel/drivers/block/loop.ko`.

Il comando: `rmmmod nome_modulo` è il modo più semplice per rimuovere un modulo. Vi si possono aggiungere, fra l'altro, l'opzione `-f`, che forza la rimozione (e che funziona solo se è stata selezionata, nella sezione del kernel `Loadable module support`, la voce: `Forced module unloading`), e l'opzione `-w`, che aspetta che il modulo non sia in uso presso nessuno degli utenti loggati nel sistema.

Il comando `modprobe` aggiunge o rimuove intelligentemente uno o più moduli del kernel che sono attualmente caricati nel sistema. Infatti è in grado anche di gestire le dipendenze dei moduli stessi. Esso può valersi di una serie di opzioni, fra cui ricordiamo:

- v: esegue il comando in modo verboso, cioè fornendo una serie di informazioni sul processo in corso.
- C --config: esegue il comando in base non alla configurazione di `/etc/modprobe.conf` o dei file di `/etc/modprobe.d`, ma in base a un file di configurazione specificato nel comando (col suo intero percorso).
- ignore-install oppure --ignore-remove: ignora i comandi di installazione o di rimozione, ed è utile soprattutto nel caso vi siano più moduli connessi.
- r: rimuove il modulo specificato; se il modulo dipende da un altro che non è in uso, cerca di rimuovere anche quello. A differenza del comando di inserimento, applicabile a un solo modulo, il comando di rimozione può applicarsi contemporaneamente a più moduli.
- l: specifica i moduli presenti nel kernel (non solo quelli attualmente caricati, per consultare i quali si ricorre invece al comando `lsmod`).

Fino a Slackware 10.2, i moduli da caricare o rimuovere col kernel 2.6 erano specificati nel file `/etc/modprobe.conf`, ma ora la situazione è cambiata. Infatti, soprattutto per gestire meglio le eventuali dipendenze fra moduli, ma anche per creare alias adeguati o caricare più moduli, si è preferito ricorrere a una directory, e cioè `/etc/modprobe.d`, in cui i vari moduli e il loro comportamento è specificato separatamente, ciascuno in un suo file. In questi file va inserito un comando per riga, mentre una barra inversa a fine riga precisa che il comando continua in quella successiva e il simbolo del cancelletto imposta un commento.

Le opzioni che possono essere precisate sono innanzitutto gli alias per un determinato modulo. Ad esempio, la stringa `alias mio_modulo modulo_con_un_nome_lunghissimo` significa che si potrà caricare quel modulo usando la forma abbreviata. Per gestire più moduli si possono adottare anche i caratteri jolly. In secondo luogo si possono aggiungere ai moduli le opzioni del caso.

Una funzione importante dei file di `/etc/modprobe.d` è che i moduli possono essere caricati non semplicemente dal kernel, ma ricorrendo a un comando specificato. Ciò si rivela utile, ad esempio, nel caso in cui un modulo lavora meglio se caricato insieme a un altro modulo già installato (e non caricato), ma che

non dipende da esso, per cui il semplice comando: `modprobe nome_modulo` non caricherebbe il secondo. Il risultato potrebbe essere qualcosa come la stringa: `install modulo_1 /sbin/modprobe modulo_2; sbin modprobe --ignore install modulo_1`. L'opzione `--ignore-install` fa in modo che la seconda occorrenza di `modprobe` non causi un nuovo tentativo di eseguire il medesimo comando di installazione. Si può anche stabilire un ordine gerarchico nel caricamento dei moduli. Ad esempio, il comando: `install probe-ethernet /sbin/modprobe e100 || /sbin/modprobe eeepro100` fa sì che si cerchi di caricare dapprima il driver `e100`, e solo nell'eventualità che il comando non riesca si cerchi di caricare `eeepro100`, allorché si lancia il comando `modprobe ethernet`.

I file di `/etc/modprobe.d` possono anche rimuovere i moduli, quando al comando `modprobe` viene aggiunta l'opzione `-r`. Il comportamento è del tutto analogo al comando di caricamento. La controparte dei due comandi appena visti è infatti: `remove modulo_1 /sbin/modprobe -r --ignore-remove modulo_1 && /sbin/modprobe -r modulo_2` nel primo caso, e `remove probe-ethernet /sbin/modprobe -r eeepro100 || sbin/modprobe -r e100` nel secondo.

C'è la possibilità di riferirsi ad altri file di configurazione col comando: `include nome_file`.

Infine, c'è la possibilità di blacklistare determinati moduli, ossia di inibirne il caricamento. Il modo classico per blacklistare un modulo è inserirne il nome o l'alias in `/etc/modprobe.d/blacklist`.

Per approntare i moduli della scheda sonora ci soccorre l'utilità `alsaconf`, che dovrebbe individuare automaticamente la scheda audio del computer. È sufficiente lanciare il comando `alsaconf` e quindi acconsentire a che il programma aggiunga le righe pertinenti in uno dei due succitati file di configurazione. L'utilità `alsaconf` non riconosce la scheda nel caso il supporto a essa sia stato inserito nel kernel, ma in tale situazione non ci si vale nemmeno dei driver di Alsa.

Per ottimizzare la resa della scheda audio, innanzitutto abilitiamo da console il server sonoro, che di default potrebbe essere muto, col comando `alsamixer`, che apre un menu pseudo-grafico in cui dobbiamo impostare i volumi delle varie applicazioni mediante i tasti freccetta; premendo il tasto a interruttore **m** si possono abilitare o disabilitare i singoli canali, naturalmente diversi a seconda del modello e delle caratteristiche della scheda audio. Un buon sistema empirico è quello di tenere gli altoparlanti del computer accesi a un volume relativamente sostenuto per poter rilevare eventuali disturbi, come fischi e fruscii, o magari suonare un CD con KSCd o qualche mp3, dopo di che si esce dal programma semplicemente premendo **Esc**. Una volta terminata l'operazione, i valori così settati vanno memorizzati col comando: `alsactl store`. Le impostazioni (uguali per tutti gli utenti) vengono trascritte nel file `/etc/asound.state`. Ulteriori modifiche possono essere apportate in un secondo momento, richiamando il mixer. Anche in questo caso i nuovi valori andranno registrati col comando: `alsactl store`.

Non resta che andare nel Centro di controllo di KDE, scegliere ALSA come dispositivo audio del sistema sonoro e fare gli ultimi eventuali aggiustamenti al mixer di KDE, e cioè `Kmixer`, e/o ricorrendo ad `aumix`.

Scheda n. 14

Il sistema sonoro in GNU/Linux

Il primo sistema audio di GNU/Linux è stato OSS/Free (Open Sound System), una serie di driver atti a fornire un'API (acronimo di «Applications Program[ming] Interface») per svariate piattaforme Unix che supportavano innumerevoli dispositivi audio. Tuttavia i driver di OSS non erano standardizzati.

In seguito lo sviluppo di OSS si concentrò su GNU/Linux e nacque così OSS/Linux, il quale tuttavia procedeva piuttosto a rilento rispetto agli sviluppi del kernel. Il lavoro degli sviluppatori finì per abbandonare progressivamente OSS, invece di rinnovarlo, a favore di un progetto del tutto nuovo: ALSA (acronimo di «Advanced Linux Sound Architecture»).

ALSA è un sistema modulare con supporto MIDI, in grado di sostituire tutte le applicazioni basate su OSS (con il quale è pienamente compatibile) in maniera anche più efficiente. Include inoltre un'API e una libreria in user-space (`libasound`) per i programmatori che intendono gestire le caratteristiche e le potenzialità del server sonoro. ALSA ricorre a `devfs` e `procfs`, astraendo dall'interazione fra hardware e software, su cui OSS era invece largamente curvato. Si appoggia pertanto a una serie di dispositivi virtuali, contenuti in `/dev/snd`. In tal modo ha acquisito una struttura organica che riesce a gestire una grande quantità di schede audio anche professionali e con supporto MIDI. L'elenco delle schede sonore supportate da ALSA si può leggere nel sito ufficiale, all'indirizzo: <http://www.alsa-project.org>. Infine, ALSA offre una serie di piccole applicazioni che consentono di controllare i vari aspetti del sistema, fra le quali: `amixer`, per regolare il volume e i controlli del suono; `alsamixer`, una versione pseudo-grafica di `amixer` basata su `ncurses`; `aconect` e `aseqnet` per connettersi a dispositivi MIDI e controllare le porte connesse; `aplay`, `aplaymidi` e `arecordmidi`, applicazioni da riga di comando per eseguire e registrare numerosi tipi di file sonori.

ALSA interagisce con dei software chiamati «server sonori» che integrano le potenzialità di quanto offerto dal kernel. Le applicazioni potrebbero usare direttamente ALSA, ma diverse schede audio (soprattutto i modelli più economici) non hanno un mixer hardware integrato (o, in certi casi, ALSA non è in grado di usarlo); di conseguenza, appena un'applicazione iniziasse a riprodurre i suoni, le altre non riuscirebbero a usarla. È per ovviare a questo limite che si è fatto ricorso, per l'appunto, ai demoni. Grazie a essi le applicazioni gestite da KDE, anziché usare l'API di ALSA, fanno ricorso a quella di **aRTS**, che si occupa di mixare tutti i flussi sonori in ingresso e di farli riprodurre dalla scheda audio. Peraltro, dati i notori limiti di aRTS e stando agli attuali annunci, esso verrà abbandonato in KDE 4 a favore di una nuova struttura multimediale chiamata Phonon, capace di usare tutti i backend audio.

Tra gli innumerevoli server sonori, oltre al già menzionato **aRTS** (il server sonoro di KDE), facilmente configurabile, ma dotato di qualità non professionale, menzioniamo **ESD** (acronimo di «Enlightened Sound Daemon», il server sonoro adottato da Gnome), con poche opzioni da configurare; **JACK** (acronimo di «Jack Audio Connection Kit»), che richiede ALSA potenziandone notevolmente le capacità. Esso funge da tessuto connettivo per tutte le applicazioni audio installate nel sistema che lo supportano e consente a un segnale audio di fluire da una all'altra di queste applicazioni per esserne modificato.

La libreria **xine-lib** permette di decodificare e riprodurre flussi audio e video mediante la sua API. Viene utilizzata da diverse applicazioni (ad esempio, da amarok) alla stessa stregua di aRTS, e – a sua volta – può utilizzare aRTS per effettuare l'output.

Gstreamer è un framework (in via di sviluppo) di flussi multimediali assai complesso: esso permette alle applicazioni di condividere un set comune di plugin per operazioni quali codifica e decodifica video, codifica e decodifica audio, filtri audio e video, visualizzazione audio, Web streaming e qualsiasi altra che fluisca in tempo reale o in altro modo. Esso presenta un'API ben definita, attraverso la quale è possibile istanziare i vari componenti e interconnetterli, creando una catena che può svolgere i compiti più disparati. In questo modo, solo Gstreamer svolge il compito di interpretare i vari formati di file e, in ultima istanza, definisce i vari formati supportati o meno da un programma, a seconda dei supporti con cui è stato compilato. Ad esempio, se compilato col supporto `xvid`, qualunque player video che usa Gstreamer sarà in grado di riprodurre file `xvid`; questa modularità spiega perché le distro che lo adottano in genere hanno un pacchetto base per `gstreamer` e altri pacchetti come `gstreamer-mp3`, `gstreamer-vorbis`, ecc.; quando nuovi formati multimediali saranno inventati, basterà inserire il supporto a questi in Gstreamer perché tutte le applicazioni, anche quelle scritte prima, siano grado di riprodurli. In conclusione, potrebbe trattarsi di un mezzo per unificare il lavoro dei maggiori Desktop Environment.

In GNU/Linux è così possibile non solo ascoltare la maggior parte dei formati audio, ma anche produrre e gestire gli effetti digitali con **LADSPA** (acronimo di «Linux Audio Developers Simple Plugin Architecture»), e registrare e mixare con una ormai nutrita serie di programmi di livello professionale. Insomma, il settore del sonoro nell'ambito dei sistemi GNU/Linux, che pure non erano storicamente nati con l'obiettivo di gestire la multimedialità, appare attualmente in piena espansione e tutto sommato

ormai soddisfacente per diverse esigenze.

6.2 Il server sonoro e gli utenti

In questo modo tuttavia il server sonoro funziona correttamente solo per l'utente root. Se vogliamo che esso funzioni correttamente anche per tutti gli altri utenti che aggiungeremo, abbiamo a nostra disposizione varie strategie. Con il kernel 2.4 possiamo modificare alcuni permessi col comando: `chmod 666` per i seguenti dispositivi:

```
/dev/snd/controlC0  
/dev/snd/hwC0D0  
/dev/snd/midiC0D0  
/dev/snd/pcmC0D0c  
/dev/snd/pcmC0D0p  
/dev/snd/seq  
/dev/snd/timer.
```

Naturalmente, in questo modo si modificano i permessi di default, per cui questo metodo andrebbe applicato solo in presenza di fondate ragioni. Inoltre, se si adotta il kernel 2.6 con `udev`, sarebbe inutile modificare i permessi, perché le modifiche sparirebbero a un successivo riavvio.

In alternativa, è possibile ottenere lo stesso risultato (indipendentemente dal kernel usato: col 2.6 i dispositivi sono infatti differenti) in modo più lineare ed elegante inserendo i nomi degli utenti nel gruppo `audio` e nel gruppo `cdrom` (quest'ultimo passaggio non è strettamente collegato all'audio in generale, quanto piuttosto alla possibilità di ascoltare CD musicali). Ciò si può fare o aggiungendo manualmente i nomi nel file `/etc/group`, oppure da console con `usermod` o con `gpasswd`. Nel primo caso daremo il comando:

```
usermod -G cdrom,audio nome_utente.
```

Questo comando presenta però uno svantaggio: se l'utente è già membro di un gruppo che non viene elencato, esso viene rimosso da tale gruppo. Pertanto, per esemplificare, se l'utente `samiel` fa già parte del gruppo `scanner`, il comando sopra riportato lo rimuoverà da quel gruppo. A questo punto, per aggiungere l'utente in questione ai due nuovi gruppi `audio` e `cdrom` senza eliminarlo dall'altro, si dovranno menzionare con `usermod` anche tutti gli altri gruppi di cui egli è già membro, per cui il comando risulterà:

```
usermod -G cdrom,audio,scanner samiel
```

È allora più comodo ricorrere al secondo comando, che assumerà la forma:

```
gpasswd -a nome_utente cdrom,audio.
```

6.3 Ascoltare i file MIDI

6.3.1 Che cosa sono i file MIDI

I file MIDI (acronimo di «Musical Instrument Device Interface») richiedono qualche accortezza in più per essere ascoltati. Essi non sono propriamente dei file sonori contenenti i suoni campionati in forma digitale (come gli `.wav`), ma piuttosto dei file dove è registrata la procedura per produrre i suoni stessi. Essi pertanto hanno dimensioni più ridotte degli altri file sonori, ma richiedono un sintetizzatore MIDI, che può essere un dispositivo hardware o software. Tale dispositivo ricorre a un certo numero di campioni sonori di strumenti, per ricostruire il brano musicale, raccolti in un file chiamato «soundfont». I soundfont sono file molto ampi, da alcuni MB ad alcune decine di MB, e vale la regola empirica che maggiori sono le loro dimensioni, migliori sono i risultati della sintesi sonora; essi infatti contengono dei

campioni di suoni digitali (ad esempio, una tromba, un violino, un pianoforte) e le istruzioni perché il sintetizzatore li possa gestire (variando inoltre, a seconda delle impostazioni, il tocco, la dinamica, effetti come il vibrato ecc.). Il file (di estensione `.fs2`, anche se esiste un altro formato, `.GUS`) che contiene questa campionatura si chiama «banco soundfont», mentre i vari campioni sonori presenti al suo interno hanno estensione `.pat` (e sono chiamati «patch»). Di consueto, un banco soundfont contiene un set di strumenti nel banco 0, alle posizioni 0-127, e quindi un set di percussioni al banco 128, posizione 0. È possibile scaricare dalla rete parecchi soundfont, e inoltre esistono delle utilità per crearne di nuovi, il che tuttavia presuppone che si faccia dei MIDI un uso abbastanza professionale. Fra i soundfont disponibili in rete, una qualità particolarmente buona va riconosciuta alla raccolta nota col nome di Eawpatches.

Fra le schede sonore non integrate, soltanto SoundBlaster ha un sintetizzatore, che però va configurato; la prima scheda ad avere il chip è stata l'ormai vecchia Sound Blaster AWE 32/64. Le schede sonore integrate invece non lo hanno mai, e allora bisogna ricorrere a un software di emulazione. È facile rendersene conto verificando (a maggior ragione in occasione di un nuovo acquisto) le schede supportate da Alsa, all'indirizzo: <http://www.linuxfocus.org/common/src/article259/SOUNDCARDS.html>.

6.3.2 Ascoltare i MIDI con schede dotate di sintetizzatore MIDI

Le varie schede SoundBlaster, come abbiamo visto, supportano nativamente i MIDI. È tuttavia necessario in primo luogo avere installato il soundfont e, secondariamente, caricarlo con un'apposita utilità. Il soundfont si trova generalmente nel CD di installazione della stessa scheda. Esso va copiato in una directory a piacere: qui consideriamo di copiarlo in `/usr/lib/sfbank` (oppure in `/etc/midi` o ancora in `/usr/share/midi`).

A questo punto il soundfont va caricato, e richiede un'apposita utilità. Essa si trova nel pacchetto `awesfx`, reperibile all'indirizzo: <http://www.alsa-project.org/~iwai/awedrv.html>. Per OSS l'utilità è `sfxload`, mentre per ALSA è `asfxload`. Manualmente, è possibile caricare il soundfont da shell, indicando la directory del soundfont, col comando:

```
sfxload /usr/lib/sfbank/nome_soundfont.SF2
```

oppure:

```
asfxload /usr/lib/sfbank/nome_soundfont.SF2
```

In alternativa, si può disporre il caricamento automatico (questa è solo una delle possibili modalità) inserendo in `/etc/rc.d/rc.local` la riga:

```
sfxload /usr/lib/sfbank/nome_soundfont.SF2
```

oppure:

```
asfxload /usr/lib/sfbank/nome_soundfont.SF2
```

È inoltre possibile personalizzare la resa, inserendo campi appropriati in un file di configurazione chiamato `/etc/sfxloadrc`, con le opzioni da attivare di default (come il path o i path in cui cercare i soundfont) o impostazioni specifiche per ogni soundfont.

A questo punto basta utilizzare un player per MIDI, come **pmidi** (un programma a riga di comando non incluso in Slackware, la cui homepage è: <http://www.parabola.me.uk/alsa/pmidi.html>, scaricabile all'indirizzo: <http://freshmeat.net/projects/pmidi>), oppure **Kmid**, presente in KDE; in quest'ultimo caso l'unica accortezza da avere è impostare correttamente il dispositivo MIDI da utilizzare. Infatti, chi usa KDE deve andare in **Centro di controllo – Suono e multimedia – Sistema sonoro – Hardware**. In quest'ultima linguetta va selezionato **Architettura Sonora Linux Avanzata (ALSA)** e nella casella **Selezione il tuo dispositivo midi** va selezionato il dispositivo MIDI. Nel caso della maggior parte delle schede SoundBlaster dovrebbe risultare:

```
Emu10K1 Port 0 - Alsa Device.
```

Pmidi è invece in grado di fornirci alcune informazioni sulle porte midi, se lanciato con l'opzione -l. Ad esempio, si potrebbe avere questo output:

```
bash-3.00# pmidi -l
Port Client name          Port name
64:0 EMU10K1 MPU-401 (UART) EMU10K1 MPU-401 (UART)
65:0 Emu10k1 WaveTable     Emu10k1 Port 0
65:1 Emu10k1 WaveTable     Emu10k1 Port 1
65:2 Emu10k1 WaveTable     Emu10k1 Port 2
65:3 Emu10k1 WaveTable     Emu10k1 Port 3
```

6.3.3 Ascoltare i MIDI con schede prive di sintetizzatore MIDI

Le schede che non sono dotate di un chip MIDI devono ricorrere a un emulatore. Il più diffuso è, attualmente, **TiMidity++**, il cui primato è però insidiato da un altro ottimo programma, **Fluidsynth**. Anche questi programmi richiedono dei soundfont (in questo caso però chiamati anche «instruments»).

È possibile ascoltare i file MIDI da console o usando delle GUI, a seconda del programma prescelto.

Usare TiMidity++

TiMidity++ è un renderizzatore di file MIDI, atto a generare i file audio; qui tuttavia esso ci interessa piuttosto come sintetizzatore in real time, in grado cioè di inviare l'uscita audio e di eseguire i MIDI in tempo reale. Con TiMidity++, oltre a installare il programma, è necessario:

- a) avere a disposizione i soundfont
- b) scompattarli in una directory (a piacere) come `/usr/lib/sfbank/timidity/instruments`
- c) creare in `/usr/share/timidity/timidity.cfg` il file `timidity.cfg` che indica la posizione del banco soundfont e l'elenco delle mappature (in parecchi casi i pacchetti di soundfont contengono anche, già compilato, il file di configurazione). Tale file potrebbe essere (ne offriamo uno stralcio):

```
soundfont /usr/lib/sfbank/timidity/instruments

bank 0

0 instruments/acpiano amp=125
1 instruments/britepno amp=115
2 instruments/synpiano amp=115
3 instruments/honky
4 instruments/epiano1
.
.
.
125 instruments/helicptr
126 instruments/applause
#127 instruments/pistol
```

```
#####
# The GUS drum patches
#
# Many patches are forced to note=xx, since they are keyed to C4 or some
# other note rather than to the key they are played on in the set. It is
# simpler to change the .cfg file than edit the patches :) All the congas
# really are best keyed to note=62, those aren't typos.

drumset 0

25 instruments/snarerol keep=env amp=225 note=60
26 instruments/snap note=65
27 instruments/highq amp=50 note=65
.
.
.
# before this point, GM/GS and XG standard drumsets are different...

35 instruments/kick1
36 instruments/kick2
.
.
.
86 instruments/surdo1
87 instruments/surdo2
```

In presenza di un file di mappatura .SF2 non è necessario scompattare nulla. Basta collocarlo in una directory a piacere, come potrebbe essere /usr/lib/sfbank, e riportarne la collocazione sempre nel sopra citato file di configurazione, e cioè /usr/share/timidity/timidity.cfg, in modo che risulti, ad esempio:

```
soundfont /usr/lib/sfbank/PC51f.sf2.
```

Un comando classico per ascoltare un file MIDI è: `timidity -iA -0s file.mid`, dove `-iA` indica di usare il dispositivo pcm di ALSA e `-0s` di usare l'interfaccia ALSA sequencer. Per lasciar fare tutto a TiMidity++, basta il comando: `timidity file.mid`.

Poiché TiMidity++ si appoggia in questo caso a /dev/snd/seq (e Kmid a /dev/sequencer, emulazione OSS del medesimo dispositivo), si deve prestare attenzione a un'eventuale compilazione del kernel 2.6 se si intende abilitare udev, poiché questi dispositivi non vengono creati automaticamente: dunque è necessario abilitare Alsa direttamente nel kernel e inserirvi le voci Sequencer support e OSS Sequencer API.

Naturalmente è possibile usare altri sistemi sonori, da Arts a Jack, modificando le opzioni. Il comando: `timidity -h` mostra una lunga lista di opzioni.

È infine possibile adottare delle GUI, se sono state abilitate nella compilazione del pacchetto. Ad esempio, lanciando `timidity -ig` si aprirà l'interfaccia grafica basata sulle gtk, e via dicendo.

Usare Fluidsynth

Con Fluidsynth il comando classico per ascoltare un file MIDI presenta la sintassi: `[comando] [opzioni] [soundfont] [file_midi]` e risulta: `fluidsynth -a alsa -m alsa_seq /usr/lib/sfbank/PC51f.sf2`

file.mid, dove *-a* abbrevia *--audio-driver*, e ancora una volta le opzioni richiedono al programma di usare il dispositivo pcm di ALSA e l'interfaccia ALSA sequencer. Più semplicemente, il comando potrebbe allora essere: `fluidsynth /usr/lib/sfbank/PC51f.sf2 file.mid`.

Usare Kmid

Per usare un'interfaccia come Kmid, è necessario lanciare come demone (cioè come ALSA sequencer) un emulatore in background (TiMidity++ ammette questa possibilità) e quindi scegliere la porta che sarà visibile nel menu delle impostazioni di Kmid. Il comando da console sarà allora: `timidity -iA -0s &` oppure: `timidity -iA -0s --background` oppure ancora, nel modo più semplice: `timidity &`.

A questo punto, nella finestra delle impostazioni di Kmid si troveranno le voci relative alle porte di TiMidity++ per il sequencer che si possono utilizzare.

Per evitare di caricare manualmente TiMidity++ in background ogni volta, è possibile aggiungere in `/etc/rc.d/rc.local` le righe:

```
# starting Timidity
timidity=/usr/bin/timidity
$timidity -iA -0s > /dev/null &
```

dove il reindirizzamento a `/dev/null` serve a evitare di ricevere, in fase di caricamento del sistema, messaggi noiosi.

Scheda n. 15

Che cosa sono i permessi

I tre tipi di permessi

In GNU/Linux, a differenza di quanto avviene in Windows, ogni applicazione presenta determinati permessi, e precisamente tre: lettura, scrittura ed esecuzione. Essi vengono indicati con lettere o con numeri: il permesso di lettura con *r* o con 4, di scrittura con *w* o con 2, di esecuzione con *x* o con 1. I permessi complessivi vengono indicati con una triade di lettere o con un numero che è la somma dei vari permessi numerici. Ad esempio, poiché root ha tutti i permessi, nei confronti di una certa applicazione avrà il permesso *rwX* (e cioè di lettura, scrittura ed esecuzione) ossia il permesso 7 (4+2+1). Se un utente ha solo il permesso di lettura, avrà il permesso *r--*, ossia 4, e via dicendo.

Inoltre i permessi sono distinti in tre triadi: la prima riguarda il proprietario dell'applicazione, la seconda il gruppo, la terza tutti gli altri utenti. Le tre triadi sono indicate sempre con lettere o numeri. Di conseguenza, se tutte le tre categorie (proprietario, gruppo, altri utenti) hanno tutti i permessi, l'applicazione avrà i seguenti permessi: *rwXrwXrwX*, e cioè 777. Se solo root ha tutti i permessi e le altre due categorie solo il permesso di lettura, avremo: *rwXr--r--*, e cioè 744, e via dicendo. I permessi di un file possono essere visualizzati da console col comando: `ls -l nome_file`

I permessi delle directory

Una situazione leggermente diversa si propone per i permessi delle directory. Il permesso *r* consente la visualizzazione del loro contenuto, il permesso *w* ne consente la scrittura e cioè la modifica consistente nell'aggiunta o nella rimozione di file, infine il permesso *x* ne consente l'attraversamento. Se si toglie quest'ultimo permesso, non sarà più possibile entrare nelle directory interessate per leggerne il contenuto. Invece, se si toglie a una directory il permesso di lettura, ma si lascia quello di attraversamento, l'utente non potrà visualizzarne il contenuto, ma, se conosce esattamente il percorso e il nome di un file presente al suo interno, potrà comunque leggerlo.

Modifica dei permessi con chmod

I permessi possono essere modificati da root col comando `chmod` (abbreviazione di «change mode»), aggiungendo la notazione a numero oppure a lettere e facendola precedere dal segno + per concedere il permesso, dal segno - per revocarlo, dal segno = per precisare che quelli e solo quelli sono i permessi concessi al file, e quindi da un altro simbolo per definire il destinatario dell'operazione. I simboli sono: a che significa «all», e cioè tutti, u che significa «user», g che significa «group» e o che significa «others», cioè tutti gli utenti non inclusi nel gruppo. Ad esempio, se root volesse rendere eseguibile un file da parte di tutti, dovrebbe dare il seguente comando: `chmod a+x nome_file`; se invece volesse revocare a tutti utenti tranne che a se stesso il permesso di leggere un file, dovrebbe lanciare il comando: `chmod go-r nome_file`; se infine volesse revocare a tutti gli utenti tranne che al proprietario il permesso di disporre in alcun modo del file, dovrebbe lanciare il comando: `chmod go-rwx nome_file`. E così via.

Casi particolari

È possibile valersi dei permessi per casi particolari. Il comando: `chmod go-w nome_file` rende il file di sola lettura per tutti, tranne che per il proprietario, mentre il comando: `chmod go= nome_file` rende il file privato, cioè inaccessibile a tutti tranne che al proprietario. Al contrario, per rendere pubblico un file, cioè leggibile e scrivibile da tutti, il comando sarà: `chmod a+rw nome_file`.

Modifica del proprietario con chown

È infine possibile cambiare il proprietario di un file col comando `chown` (che abbrevia «change owner»), con la seguente sintassi: `chown nome_nuovo_proprietario nome_file`.

Si possono cambiare contemporaneamente proprietario e gruppo indicandoli di seguito, separati dai due punti, con il seguente comando: `chown nome_proprietario:nome_gruppo nome_file`.

Questi comandi supportano l'opzione `-R`, che modifica ricorsivamente permessi e proprietari anche delle subdirectory.

Umask

Quando si ha a che fare con sistemi diversi da GNU/Linux, che non presentano perciò in forma nativa i permessi (come accade con Windows), si è costretti a modificare l'accesso ai file in altro modo, e cioè intervenendo in `/etc/fstab` con `umask`. Da un punto di vista tecnico, `umask` si definisce il complemento ottale di `chmod`, in quanto viene formulato numericamente come `chmod`, ma funziona alla rovescia. Qui la concessione di tutti i permessi è espressa dalla formula numerica `000`, la negazione di qualsiasi permesso dalla formula `777`, mentre la concessione del permesso di lettura e di esecuzione (le directory devono essere sempre dotate del permesso di eseguibilità, che non a caso viene chiamato «attraversamento», perché gli utenti possano accedervi), ma non di scrittura al gruppo e agli utenti sarà `022` (complementare di `chmod 755`), e così via. Nei file system non nativi, `umask` non si può differenziare file per file, mentre i permessi, in GNU/Linux, possono essere diversi per ciascun file.

Ogni nuovo file ha un `umask` di default, che si può leggere in `/etc/profile` alla sezione `Default umask` o che si può richiamare da shell col comando `umask`, e che in Slackware è pari a `022`. Ciò significa che root ha il permesso 7, mentre gruppi e utenti hanno il permesso 5, e cioè di lettura ed esecuzione, ma non di scrittura. È possibile modificare questo settaggio editandolo manualmente.

I permessi suid e sgid

Un permesso speciale è il cosiddetto **bit suid** (set-user-ID bit, abbreviato in **setuid**). Un altro permesso speciale è il cosiddetto **bit sgid** (set-group-ID bit, abbreviato in **setgid**).

Quando si lancia un eseguibile, il kernel ne imposta di norma il proprietario e/o il gruppo del processo in corrispondenza a quelli dell'utente e del gruppo con cui lo si lancia. Se però questo eseguibile ha il bit suid impostato (lo stesso vale per sgid), il kernel assegna al processo l'uid del proprietario (o del gruppo) e non dell'utente che lo ha avviato. In questo modo si consente agli utenti normali di usare programmi che richiederebbero permessi speciali. Fra i casi più noti, menzioniamo il comando `passwd`, che modifica il file delle password e che può essere scritto solo da root. Poiché tuttavia anche gli utenti

devono poter essere in grado di cambiare la loro password da soli, senza richiedere l'intervento dell'amministratore del sistema, il comando in questione appartiene a root e ha il bit `suid` impostato; quando esso è invocato da un utente normale, parte comunque con privilegi di root.

L'impostazione di questi permessi dev'essere attentamente valutata, perché può introdurre falle nella sicurezza del sistema. I bit `suid` e `sgid` si impostano coi comandi `chmod` e `chgrp` con l'opzione `+s`, e possono essere verificati col comando: `ls -l`, che visualizza una `s` al posto della `x` in corrispondenza dei permessi di utente o gruppo.

Da notare che, se un file ha il bit `suid/sgid`, ma non quello di esecuzione, la `S` è maiuscola. Se invece è una directory ad avere il bit `sgid` impostato, tutti i file creati al suo interno avranno i suoi stessi attributi (proprietario e gruppo), indipendentemente da chi li crea, e tutte le directory create al suo interno saranno dotate, ricorsivamente, del bit `sgid`.

La concessione dei permessi con sudo

La concessione di permessi speciali si può effettuare con `sudo` (acronimo di «superuser do»). Vi sono infatti delle operazioni riservate a root, che però può decidere di farle compiere anche a determinati utenti. In questo caso egli concede loro, in forma perpetua o revocabile, il permesso di svolgerle. A tal fine è necessario editare il file `/etc/sudoers`. Benché sia prescritto di editarlo con `visudo`, una versione di Vi studiata apposta per `/etc/sudoers` che è in grado di compiere una verifica della correttezza della sintassi del file, in realtà è possibile modificarlo con qualsiasi editor, ed effettuare in seguito il controllo lanciando il comando `visudo -c`. Infatti, se editato per intero con `visudo`, il file risulterà «lockato», ovvero sarà creato un file `.tmp` che blocca il file indicando che esso è in uso, per impedirne più accessi contemporanei. `Sudo` è in grado di concedere permessi definiti e delimitati a determinati utenti o gruppi di utenti e si serve di una sintassi propria. Rinviamo qui all'ampia pagina man del programma.

Capitolo 7

Gestione degli utenti e dei gruppi

7.1 Gestione degli utenti

GNU/Linux è strutturalmente un sistema multi-utente. Abbiamo visto i vantaggi di non lavorare da root, se non per compiti amministrativi, nella **Scheda n. 4 – Che cos'è la partizione /home**. Ciò implica che, per tutti gli usi comuni, è più che opportuno creare almeno un utente non root.

7.1.1 Creare gli utenti

La procedura di creazione di nuovi utenti è molto semplice, e naturalmente è un compito destinato soltanto a root. Da console, è sufficiente digitare il comando `adduser`, al che viene chiesto 1) il nome del nuovo utente; 2) il numero (uid = user id) da attribuirgli (i numeri uid degli utenti aggiunti iniziano da 1000, perché tutti i precedenti sono riservati agli utenti di sistema), e se non ci sono esigenze particolari va bene quello proposto di default, e cioè il primo numero libero; 3) il gruppo primario in cui collocarlo (di default è `users`); 4) gli altri eventuali gruppi in cui collocarlo (che vanno indicati di seguito, separati da virgole senza spazi intermedi); 5) la sua home; 6) la sua shell (di default in Slackware si tratta di `bash`); 7) la scadenza (e cioè il tempo per il quale quell'account sarà disponibile, dopo di che verrà automaticamente disattivato); 8) altre eventuali informazioni personali; infine 9) la password.

L'utente ha accesso solo a una parte dei programmi, normalmente quelli contenuti in `/bin`, `/opt`, `/usr`, `/usr` e relative subdirectory. Root può modificare i permessi (e creare gruppi) per consentire o impedire agli utenti di usare un determinato programma.

Fra i comandi aggiuntivi più utili annoveriamo `passwd`, che permette di cambiare la password di un utente, `chsh`, che cambia la shell predefinita dell'utente (come sappiamo, `bash`) e infine `chfn`, che cambia le informazioni di `finger` (dette anche – sebbene l'espressione sia ormai obsoleta – GECOS, acronimo di «General Electric Comprehensive Operating Supervisor». Questo campo, un tempo adottato da Bell Labs nell'implementazione originale di UNIX per i suoi diversi computer, è stato usato per conservare informazioni quando il sistema UNIX inviava batch e lavori di stampa al sistema GECOS), e cioè quelle, aggiuntive e del tutto opzionali, sull'utente (nome reale completo, indirizzo, numero telefonico, ecc.). Per le opzioni di questi comandi, rimandiamo alle rispettive pagine `man`.

7.1.2 Rimuovere gli utenti

La rimozione di un utente è effettuata da root col comando `userdel nome_utente`. Esso, inoltre, modifica alcuni file che possono essere editi anche manualmente: `/etc/passwd`, dove viene rimossa la riga corrispondente all'account da eliminare; `/etc/shadow`, dove allo stesso modo viene rimossa la riga

corrispondente all'account da eliminare; `/etc/group`, dove viene rimosso l'utente da tutti i gruppi in cui è presente; viene infine eliminato l'eventuale file di spool della sua mail in `/var/spool/mail/utente`. Il comando `userdel -r nome_utente` rimuove anche la home dell'utente e tutti i file in esso contenuti.

7.1.3 Disabilitare gli utenti

In alternativa è possibile, sempre da parte di root, disabilitare un utente, e cioè impedirgli l'accesso al sistema, mantenendone però tutte le impostazioni. Ciò si effettua col comando: `passwd -l nome_utente`, dove l'opzione `-l` disabilita un account cambiando la password in un valore criptato che non coincide con nessun altro, mentre il comando `passwd -u nome_utente` serve per riabilitare l'utente, ripristinando la password iniziale.

7.1.4 Gestire le quote degli utenti

In un sistema multiutente come GNU/Linux può essere utile fissare uno spazio per i singoli utenti e per i vari gruppi, in modo che ciascuno possa disporre nel disco fisso di risorse non illimitate. Questa funzione, a dire il vero, riveste una sua utilità solo nei computer con un gran numero di utenti, ma ne indichiamo comunque le linee generali. L'attribuzione di questi spazi limitati si effettua ricorrendo alle «quote», definendo ad esempio lo spazio riservato alla posta di un singolo utente. Più precisamente, possono essere considerati due aspetti della gestione disco: il numero di inode e il numero di blocchi disponibili per un utente o un gruppo.

Il supporto per quota deve essere abilitato a livello di kernel e richiede delle utilità per la gestione e la configurazione dei limiti dello spazio da assegnare. L'abilitazione si effettua aggiungendo al quarto campo dei parametri di `/etc/fstab` la voce `usrquota` per abilitare la quota utente e `grpquota` per abilitare la quota gruppo.

Per determinare la quota di un utente sul proprio sistema, si dà il comando: `edquota -u utente`

In tal modo si apre `vi` (o comunque l'editor specificato nella propria variabile di ambiente `$VISUAL`) per modificare la quota di quell'utente su ogni partizione dove le quote sono state abilitate. Appare un buffer come il seguente:

```
Disk quotas for user samiel (uid 1001):
  Filesystem  blocks    soft   hard   inodes    soft   hard
  /dev/hda3   13046    15360  19200   806       1500   2250
```

dove «blocks» indica il numero totale di blocchi e «inodes» è il numero totale di inode (sempre in KB) che l'utente ha a sua disposizione sulla partizione. Analogamente, per determinare la quota di un gruppo sul proprio sistema, si dà il comando: `edquota -g gruppo` e allo stesso modo si aprirà `vi` per modificare le quote di questo gruppo:

```
Disk quotas for group fotografie (gid 101):
  Filesystem  blocks    soft   hard   inodes    soft   hard
  /dev/hda3   13046    15360  19200   806       1500   2250
```

Le impostazioni stabilite coi comandi `edquota -u utente` e `edquota -g gruppo` vengono registrate rispettivamente nei file `/home/aquota.user` e `/home/aquota.group`.

Il parametro **Soft (Limit)** indica l'ammontare massimo d'uso del disco di cui un utente dispone su una partizione. Quando viene combinato con Grace Period, esso agisce come una linea di confine, emettendo un avvertimento quando un utente la oltrepassa.

Il parametro **Hard (Limit)** funziona quando è impostato Grace Period. Specifica il limite assoluto d'uso del disco, che non può essere violato da un utente.

Il parametro **Grace (Period)**, lanciato col comando `edquota -t`, è il limite di tempo (possono essere usate unità di tempo di secondi, minuti, ore, giorni, settimane e mesi) disponibile prima che venga fatto rispettare il Soft Limit. Di default, in Slackware esso ammonta a sette giorni. Questo è infatti quanto appare lanciando il comando `edquota -t`:

```
Grace period before enforcing soft limits for users:
Time units may be: days, hours, minutes, or seconds
Filesystem Block grace period Inode grace period
/dev/hda3 7days 7days
```

Per visualizzare un riepilogo delle informazioni sulle quote per un file system si usa il comando `repquota -a`, che produce un output del tipo:

```
*** Report for user quotas on device /dev/hda3
Block grace time: 7days; Inode grace time: 7days

```

User	used	Block limits			File limits			
		soft	hard	grace	used	soft	hard	grace
root	-- 33792	0	0	0	85	0	0	0
samiel	-- 591492	15360	19200	7	3959	1500	2250	7

7.2 Le password

7.2.1 Il sistema delle password in Slackware

Nella gestione degli utenti, un ruolo particolarmente delicato è rappresentato dalle password. Slackware, che utilizza il sistema *shadow password*, memorizza i dati degli utenti nel file `/etc/passwd`; esso tuttavia, poiché dev'essere accessibile a tutti gli utenti, non contiene anche le password (ma solo l'indicazione, consistente nella lettera «x», che c'è una password). Una riga del file `/etc/passwd`, relativa a un fittizio utente **prova**, apparirà dunque:

```
prova:x:1001:100:,,,:/home/prova:/bin/bash
```

I vari campi, separati dal doppio punto, sono in sequenza: nome utente, password, User ID, Group ID (il gruppo degli utenti `users` in Slackware è, di default, il numero 100), eventuali informazioni opzionali per il finger separate da virgole (qui assenti), la home directory e la shell prescelta.

Le password sono invece presenti nel file `/etc/shadow`, in forma criptata. Analizziamo una riga del file `/etc/shadow`, che potrebbe apparire così:

```
prova:$1$uDl/w0z4$tMnrD0o0to76p9VWLXjkv1:12836:0:99999:7:::
```

Anche qui i vari campi sono separati dal doppio punto e, in sequenza, sono: nome utente, password criptata, giorni passati dall'ultimo cambio di password (rispetto a Epoch, fissato per ora al 1° gennaio 1970), giorni prima che la password possa essere cambiata (qui lo zero indica che la password può essere cambiata da subito), giorni che mancano allo scadere della password (il campo 99999 equivale a dire che la password non scadrà mai), giorni che mancano allo scadere dell'account dell'utente, giorni che mancano alla sospensione dell'utente, giorni dopo la sospensione per la disabilitazione dell'utente, e un campo riservato. Se non si usa la funzione di sospensione dell'account, questi ultimi campi non sono significativi.

7.2.2 Se si dimentica la password di root

Se si dimentica la password di root è impossibile accedere al sistema e pertanto si deve avviare da CD. Elenchiamo i passaggi da seguire per poter rimediare a questa situazione:

- 1) avviare il computer col CD 1 di Slackware (come per una normale installazione)
- 2) scegliere eventualmente la tastiera italiana
- 3) fare il login come root (non verrà chiesta alcuna password)
- 4) montare sulla directory /mnt la partizione di root in lettura e scrittura col comando:

```
mount -t file_system /dev/hdxn /mnt
```

dove *file_system* è appunto il tipo di file system (dunque EXT2, EXT3 o Reiserfs), *x* è il disco fisso e *n* il numero della partizione di root.

5) utilizzando chroot (comando atto a modificare la directory root per l'esecuzione di un determinato comando) entrare in /mnt come root col comando: `chroot /mnt`

6) lanciare il comando `passwd` e reimpostare una password

7) in alternativa editare /etc/shadow con un editor, ad esempio: `vi /etc/shadow` e cancellare il campo della password di root. Al successivo accesso la password non sarà richiesta, e sarà possibile reimpostarla, sempre col comando `passwd`. Con la stessa procedura si cancellerà la password di Lilo, se ne è stata impostata una.

Se non si ricorre a chroot, ma a cd, ricordarsi di richiamare /mnt/etc/shadow e non semplicemente /etc/shadow.

In questo modo una riga come:

```
prova:$1$uDl/w0z4$tMnrD0o0to76p9VWLXjkv1:12836:0:99999:7:::
```

dovrà diventare:

```
prova::12836:0:99999:7:::
```

Ancora una volta, salvato il file e riavviato il sistema, sarà possibile loggarsi come root senza ricevere alcuna richiesta della password, che potrà essere ripristinata col comando `passwd`.

7.2.3 Se si dimentica la password di utente

Dimenticare la password di utente è una faccenda meno complessa che dimenticare quella di root. Per ripristinare l'accesso dell'utente bisognerà rivolgersi a root, il quale potrà scegliere due strategie.

La prima, probabilmente la più semplice, è impostare *ex novo* una password col comando `passwd`. Infatti, mentre gli utenti per cambiare password con `passwd` devono comunque inserire quella vecchia, essa non è richiesta a root. Questi imposterà una nuova password per l'utente e (se si tratta di due

persone diverse) gliela comunicherà. A questo punto l'utente potrà effettuare l'accesso e cambiarla a suo piacimento.

La seconda strategia consiste nel cancellare la password dell'utente in `/etc/shadow`. Di conseguenza, la riga:

```
prova:$1$uDl/w0z4$tMnrD0o0to76p9VWLXjkv1:12836:0:99999:7:::
```

dovrà diventare:

```
prova::12836:0:99999:7:::
```

Al nuovo accesso, possibile a questo punto senza nessuna password, l'utente potrà reimpostarla. In questo modo però, se il computer è condiviso da più persone, si potrebbe lasciare l'accesso all'account dell'utente del tutto incontrollato finché questi non reimposti la sua password.

7.3 Gestione dei gruppi

7.3.1 I gruppi

Root può decidere che solo alcuni utenti sono autorizzati a utilizzare determinati programmi o ad accedere a determinati file o directory. Questi utenti vengono così raccolti in gruppi a cui root attribuisce un nome a piacere. In realtà, in Slackware esistono già dei gruppi predefiniti, elencati nel file `/etc/group`.

7.3.2 Creare un nuovo gruppo

Si crea un nuovo gruppo col comando: `groupadd nome_gruppo`. Si elimina un gruppo col comando: `groupdel nome_gruppo`. Infine, si possono modificare alcune caratteristiche del gruppo, come il nome o il GID (acronimo di Group Identifier, e cioè il numero che lo identifica) col comando `groupmod`. Così, per modificare il nome di un gruppo, si avrà la seguente sintassi: `groupmod vecchio_nome nuovo_nome`.

7.3.3 Aggiungere utenti a un gruppo

L'aggiunta di un utente a un gruppo è possibile manualmente, cioè editando opportunamente il file `/etc/group`, oppure col comando: `usermod -G nome_gruppo nome_utente`.

Per effettuare questa operazione, l'utente in questione non dev'essere connesso al sistema e il gruppo deve già esistere.

Infine, se si vuole aggiungere contestualmente un utente a più gruppi, i loro nomi vanno elencati di seguito, separati da virgole senza spazi intermedi; ad esempio: `usermod -G audio,burning,archivio samiel`

L'accesso a file o directory per un gruppo si può stabilire in `/etc/fstab`, mentre la possibilità di utilizzare determinati programmi si concede modificando il proprietario con il comando `chown` oppure attribuendo a un gruppo creato apposta la possibilità di utilizzarli.

Capitolo 8

Gestione delle unità

8.1 I dispositivi in GNU/Linux

8.1.1 Aggiungere dispositivi

Per aggiungere unità come partizioni del disco rigido, masterizzatori, lettori CD o DVD, pendrive, che Slackware non riconosce in automatico, è necessario suggerire a `/etc/fstab` la loro esistenza e alcune loro caratteristiche.

8.1.2 I tipi di dispositivo

In GNU/Linux i vari dispositivi vengono identificati sempre con dei file, che in genere sono raccolti nella directory `/dev` (= devices). La prima distinzione da fare è quella fra **dispositivi a caratteri** e **dispositivi a blocchi**. Trascurando altre differenze tecniche (come il carattere sincrono o asincrono del comportamento), i dispositivi a blocchi operano trasferendo blocchi di dati di lunghezza prefissata (in genere 512 byte), i quali sono indirizzabili e quindi accessibili singolarmente. Un tipico dispositivo a blocchi è costituito da un disco fisso. Nei dispositivi a caratteri è invece possibile compiere operazioni di I/O relativi a un numero arbitrario di byte. Ad esempio, il kernel invia un certo numero di dati a una stampante, dopo di che la responsabilità di gestire questi dati spetta esclusivamente alla stampante, mentre il kernel non può riesaminarne il flusso.

Dispositivi a caratteri:

le console = `/dev/tty[n]`

le porte seriali = `/dev/ttyS[n]`

le stampanti (porte parallele) = `/dev/lp[n]`

Dispositivi a blocchi:

dischi rigidi IDE = `/dev/hda`, `/dev/hdb` ecc.

dischi rigidi SCSI = `/dev/sda`, `/dev/sdb` ecc.

cd-rom IDE = `/dev/hdb`, `/dev/hdc`, `/dev/hdd` ecc.

cd-rom SCSI = `/dev/srn`, `/dev/sgn` ecc.

floppy = `/dev/fd0` ecc.

e inoltre masterizzatori, pendrive, dischi zip ecc. L'attribuzione dei nomi dei dispositivi consegue alla loro collocazione fisica. Se in un computer è presente un solo disco fisso e un lettore CD, collocati alla prima porta IDE come master e slave, essi prenderanno i nomi di `/dev/hda` e `/dev/hdb`. Se gli stessi invece fossero connessi entrambi come master alle due porte IDE, risulterebbero `/dev/hda` e `/dev/hdc`, e così via.

Esistono in aggiunta i **dispositivi a pipe**, analoghi ai dispositivi a caratteri, che però concludono il flusso di I/O con un altro processo, che può essere rediretto. Ad esempio, GPM può duplicare gli

eventi del mouse e inviarli al pipe `/dev/gpmdata` per farli utilizzare da altri programmi. Infine ci sono i **dispositivi a socket**, speciali descrittori di file che offrono un'interfaccia di rete.

8.1.3 Creare nuovi dispositivi

Col kernel 2.4, eventuali nuovi dispositivi possono essere creati singolarmente in `/dev` col comando `mknod`, con la seguente sintassi:

```
mknod -m 660 /dev/nome_dispositivo b | c | u numero_primario numero_secondario
```

dove 660 (o altro) sono i permessi, *nome_dispositivo* può essere `/dev/cdrom` o `/dev/sda1`, `b` indica un dispositivo a blocchi, `c` indica un dispositivo a caratteri e `u` indica un dispositivo senza buffer. I numeri major (primari) e minor (secondari) indicano il tipo di dispositivo e vengono indicati nel file `/usr/src/linux/Documentation/devices.txt`. Ad esempio, per creare il dispositivo `/dev/hdd` il comando sarà: `mknod -m 660 /dev/hdd b 22 64`.

È anche possibile (ri)creare tutti i dispositivi di `/dev` con un unico comando: `MAKEDEV` (che è uno script di shell presente in `/dev`), del tutto innocuo se le periferiche esistono già nel sistema.

Il ricorso ai numeri primario (*major number*) e secondario (*minor number*) serve alla gestione dei dispositivi. L'invocazione a una determinata funzione di un dispositivo (detta «system call»), che può essere `open()`, `close()`, `read()`, `write()` ecc., per poter essere effettuata invoca il driver pertinente ricorrendo al numero primario che la identifica. Tutti i dispositivi dello stesso tipo hanno però lo stesso numero primario. Ecco allora che il numero secondario serve a identificare, fra i dispositivi dello stesso tipo, quello specifico che è stato richiamato. Il numero primario che identifica i dispositivi di tipo joystick è il 15, ma poi il primo joystick analogico avrà come numero secondario 0, il secondo 1 e così via. In tal modo i vari dispositivi vengono definiti in maniera assolutamente univoca.

8.2 Devfs e udev

8.2.1 Il kernel 2.4 e devfs

Nel kernel 2.4 la gestione dei file di dispositivo è amministrata in modo statico in base a quelli presenti in `/dev`, che sono creati in fase di installazione. La directory `/dev` prevede dunque tutti i dispositivi presenti nel sistema o che non vi fanno parte, ed è perciò estremamente affollata. È inoltre possibile crearne di nuovi, come abbiamo visto, col comando `mknod`.

Il kernel 2.4 consente anche la gestione automatica dei dispositivi mediante `devfs` (acronimo di DE-Vice File System). Nella modalità predefinita (oppure mediante l'opzione di avvio `devfs=mount`), nella directory `/dev` viene montato un file system virtuale che contiene tutti i dispositivi effettivamente riconosciuti dal kernel. Tale sistema risulta più articolato rispetto a quello determinato dalla gestione tradizionale: ad esempio, il file dispositivo che si riferisce al primo disco fisso IDE è tradizionalmente `/dev/hda`, mentre con `devfs` risulta essere `/dev/ide/host0/bus0/target0/lun0/disk`. Poiché il file system creato da `devfs` è virtuale, si generano due tipi di problemi: nel caso si modificano i permessi oppure si creino dei collegamenti, entrambi vengono meno quando si spegne il sistema. Per ovviare a questo problema, il demone `devfsd` dev'essere configurato in modo da ricostruire la situazione desiderata a ogni avvio del sistema. Inoltre `devfs` presenta alcuni problemi architetturali che ne hanno determinato l'abbandono definitivo in favore di un nuovo sistema: `udev`.

In Slackware 11.0 il pacchetto `devs` non va rimosso nemmeno nel caso si adotti un kernel 2.6 con `udev`, perché è ancora richiesto durante l'avvio del sistema.

8.2.2 Il kernel 2.6 e udev

Con il kernel 2.6 la gestione dei dispositivi è stata profondamente modificata a vantaggio esclusivo di un sistema dinamico che crea i dispositivi stessi quando un programma li richiama e li abolisce quando non sono più necessari, e che è affidato non più a devfs (ritenuto ormai obsoleto), bensì a udev. Udev si comporta in maniera analoga a devfs, ma non viene gestito direttamente dal kernel, bensì da un programma esterno, il demone udevd.

In realtà, udev si compone di tre diversi progetti: **namedev**, **libsysfs** e, appunto, **udev**. I vantaggi che presenta riguardano l'assegnazione di nomi persistenti alle periferiche e l'organizzazione stabile di più periferiche rimovibili, con vantaggi anche in `/etc/fstab`.

Namedev definisce i nomi delle periferiche indipendentemente da udev; risulta pertanto molto flessibile e personalizzabile per le più diverse esigenze. Namedev controlla se le periferiche che riscontra non abbiano già un nome seguendo cinque tappe: 1) etichetta o numero seriale (ad esempio, i dispositivi USB hanno un numero seriale univoco e quelli SCSI un UUID unico); 2) numero della periferica sul bus (visto che, in genere, assai di rado si modifica la collocazione fisica di una periferica); 3) topologia del bus (cioè in base al modo in cui la periferica è collegata fisicamente al bus. La topologia cambia solo se si modifica la collocazione dell'hardware, ad esempio se si invertono i collegamenti con dischi fissi, masterizzatori ecc.); 4) nome assegnato staticamente (con una stringa arbitraria); 5) nome fornito dal kernel (è il comportamento predefinito di udev, se non trova regole specifiche per l'assegnazione di un nome). Se in uno di questi livelli namedev trova il nome, lo assume e interrompe la procedura di assegnazione.

Libsysfs interagisce col kernel attraverso sysfs. Mette a disposizione un'API (ossia un'interfaccia per accedere alle funzioni offerte) per rilevare così le informazioni fornite da sysfs. In questo modo qualsiasi dispositivo hardware può essere interrogato, prescindendo da quale sia in dettaglio.

Quando il kernel percepisce che sono stati collegati o rimossi dei dispositivi, lancia hotplug. Questi, a sua volta, lancia i programmi linkati in `/etc/hotplug.d/default`, dov'è situato anche un link simbolico (sui link simbolici, vedi Scheda n. 16 – I link simbolici) all'eseguibile udev. In tal modo è hotplug a inviare le informazioni provenienti dal kernel a udev, che si preoccupa di aggiungere o rimuovere gli opportuni file di dispositivo nella directory `/dev`, servendosi delle informazioni sui dispositivi presenti in `/sys`.

Tuttavia gli ultimi sviluppi di udev prescindono da hotplug, se si adotta il kernel 2.6. Infatti, le sue versioni più recenti effettuano le sue stesse funzioni e si configurano in user space grazie alle regole dei suoi file. Ciò consente il rilevamento dei dispositivi (peraltro con maggiore velocità rispetto a hotplug) e il loro montaggio automatico.

Slackware non si è ancora orientata in modo deciso in questa direzione. Nel Changelog del 24 aprile 2006 si legge infatti, a proposito di hotplug:

Coi kernel 2.4 si usa `/sbin/hotplug` per individuare i dispositivi, ma coi kernel 2.6 si usa `/sbin/udevsend` (se si adotta udev). Ciò dovrebbe lavorare meglio nei sistemi che hanno i kernel 2.6 con udev e HAL.

Ma Slackware ha sempre diffidato da HAL, almeno fino a questo momento. E subito dopo, a proposito di udev, anche in relazione al lavoro di Piter Punk, si legge:

Ciò che sta accadendo è molto interessante, ma ci sono delle ragioni che mi hanno spinto a fare solo dei piccoli passi in questa direzione. Ad esempio, era bello popolare `/dev` prima di verificare le partizioni e montarle in modalità di lettura e scrittura, ma sembra che ciò non sia più possibile. Inoltre, abbiamo notizia di componenti hardware che non sono stati identificati correttamente (ed è capitato anche a me). Fondamentalmente, sembra trattarsi del problema di trovare, nel processo di avvio, il momento giusto dove piazzare udev, ma ci sono anche parecchi problemi riguardante le regole di udev. Noi ci adegueremo, ma probabilmente non nella prossima versione. Questo aggiornamento a udev-071 fornisce l'adeguamento minimo alla documentazione del kernel 2.6.16.9, è stato attentamente testato e funziona bene.

udev-0.90 avvia più velocemente la macchina, ma non è altrettanto affidabile (almeno nei test compiuti qui, in base a ciò che viene invocato dai nostri script di inizializzazione), e noi non siamo mai stati dell'avviso di preferire la velocità all'affidabilità.

La versione di udev attualmente presente in Slackware, nata da una lunga ed e laborata serie di versioni, sembra ora intercettare adeguatamente gli eventi e può di conseguenza prescindere ormai da hotplug. Sarà quindi possibile disinstallare hotplug o semplicemente disabilitarlo mediante: `chmod a-x /etc/rc.d/rc.hotplug`. Tale versione richiede il kernel 2.6.15 o superiori

8.2.3 La configurazione di udev

La configurazione di udev è basata su alcuni file contenuti in `/etc/udev` e in `/lib/udev`. Il primo è `/etc/udev/udev.conf`, che precisa:

1) dov'è montato il file system di udev, e cioè dove devono essere creati i node (o collegamenti) per i vari dispositivi. Leggiamo infatti in quel file:

```
# udev_root - where in the file system to place the device nodes
udev_root=/dev/
```

Ciò significa che i dispositivi verranno montati in `/dev`.

2) quale directory contiene il database di udev:

```
# udev_db - The name and location of the udev database.
udev_db=/dev/.udev.tdb
```

Ciò significa che il database di udev risiede nel file nascosto `/dev/.udev.tdb`.

3) quale directory contiene le regole:

```
# udev_rules - The name and location of the udev rules dir
udev_rules=/etc/udev/rules
```

Ciò significa che le regole sono scritte nel file `/etc/udev/rules.d/udev.rules` e negli altri file della directory. Questo file non dovrebbe essere modificato: nuove regole possono essere aggiunte semplicemente inserendo nella stessa directory un nuovo file. Questa prassi è conveniente perché, se si aggiorna udev, `/etc/udev/rules.d/udev.rules` viene sovrascritto cancellando le regole aggiunte; certo, è sempre possibile farne un backup e poi copiare le nostre regole nel file nuovo, ma tutto questo lavoro si risparmia inserendole in un altro file, nuovo.

Tutti i file di regole creati dovranno comunque avere l'estensione `.rules`. Poiché i file di questa directory sono letti in ordine alfabetico, se si vuol far sì che le regole personalizzate siano lette prima del file di default, bisogna avere l'accortezza di chiamare il file in modo che, alfabeticamente, preceda `udev.rules`.

Nelle versioni più vecchie di udev c'era inoltre la necessità di specificare quale directory conteneva i permessi e comparivano pertanto anche le righe:

```
# udev_permissions - The name and location of the udev permission dir
udev_permissions=/etc/udev/permissions.d
```

Ciò significa che i permessi erano scritti nel file `/etc/udev/permissions.d/udev.permissions` (ma, a dire il vero, lo stesso risultato si otteneva se tali informazioni erano contenute in `/etc/udev/udev.conf`). Qui trovavamo elencati i nomi dei dispositivi, l'utente e il gruppo di appartenenza, e i permessi nella classica forma ottale. L'ultima parte di questi file specifica le proprietà dei dispositivi che non sono governati da regole particolari.

Nelle più recenti versioni sono state modificate alcune caratteristiche di udev, sia per quanto riguarda le modalità per conferire i permessi, sia per quanto riguarda la sintassi delle regole. Infatti non esiste più il file `/etc/udev/permission`, e i permessi e i mode vanno collocati in `/etc/udev/rules.d/udev.rules` o in altri file appositamente creati e collocati in questa directory.

8.2.4 La sintassi di udev

Le regole per udev seguono una sintassi comune: `key, [key, ...] name, {symlink}`.

I campi `key` e `name` sono obbligatori, `symlink` è invece opzionale (va specificato solo se un dispositivo dev'essere disponibile da più node, poiché udev crea un singolo node per ciascun dispositivo).

Il campo `key` deve indicare, e perciò coincide con, un particolare dispositivo; questo campo viene precisato nella forma `type=valore`. I diversi tipi di campo `key` sono:

BUS coincide col tipo di bus del dispositivo. Esempi tipici sono: `usb`, `scsi` e `ide`.

KERNEL coincide coi nomi assegnati dal kernel. Esempi tipici sono: `hda`, `tty`, `usb0` e `lp0`.

SUBSYSTEM coincide coi nomi assegnati dal subsystem del kernel. Generalmente, ma non sempre, tali campi coincidono col nome di bus.

DRIVER coincide coi nomi dei driver che controllano il dispositivo. Esempi sono: `radeon` e `usb-storage`.

ID coincide col numero del dispositivo del bus a cui è collegato.

PLACE coincide con la posizione fisica del dispositivo sul bus. Ciò è valido quasi esclusivamente per dispositivi USB.

SYSFS {nomefile} si tratta del nome del dispositivo che udev legge in `/sys` e per il quale cerca di individuare il valore. Può trattarsi della marca, del numero seriale, dell'UUID o dell'etichetta di sistema di un dispositivo (in ogni regola, possono essere posti fino a cinque campi di questo tipo, tra graffe).

PROGRAM questo campo consente a udev di richiamare un qualsiasi programma esterno e di usare l'output per denominare i dispositivi.

RESULT questo campo ricorre all'output finale di PROGRAM (e può dunque essere usato solo dopo un call di PROGRAM).

NAME questo campo dice a udev come chiamare il dispositivo in `/dev`. Il suo valore è il key relativo, a cui viene pertanto assegnato questo NAME.

SYMLINK è il nome di un link simbolico creato per richiamare il NAME e specifica localizzazioni addizionali dove il dispositivo sarà linkato. Si possono usare più SYMLINK separati da uno spazio. Se il link si inserisce in una regola priva di NAME, la regola non viene applicata, ma resta in attesa di una regola che crei il node. Ciò può risultare utile per creare un symlink a un node di un dispositivo, riferito a un utente specificato più tardi in un file di regole.

8.2.5 L'identificazione dei dispositivi

Per identificare i dispositivi, come abbiamo visto, si possono usare campi diversi. Il modo migliore per conoscere quale campo conviene usare in una regola sta nel ricorso al comando `udevinfo`, che si riferisce ai dispositivi elencati in `/sys`. Ad esempio, per avere informazioni sul mouse si adotta il comando: `udevinfo -a -p /sys/class/input/mouse0` e per avere quelle su una stampante USB il comando: `udevinfo -a -p /sys/class/usb/lp0`.

Udevinfo inizia listando le informazioni sul node richiamato, e prosegue cercando quelle sugli altri node dei dispositivi dello stesso tipo, elencando tutti i possibili attributi nel formato delle key di udev. Il secondo comando che abbiamo menzionato fornirà un output di questo tipo:

```
samiel@darkstar:~\$ udevinfo -a -p /sys/class/usb/lp0

udevinfo starts with the device the node belongs to and then walks up
the device chain, to print for every device found, all possibly useful
attributes in the udev key format.
Only attributes within one device section may be used together in one
rule, to match the device for which the node will be created.

device '/sys/class/usb/lp0' has major:minor 180:0
  looking at class device '/sys/class/usb/lp0':
    KERNEL=="lp0"
    SUBSYSTEM=="usb"
    SYSFS{dev}=="180:0"

follow the "device"-link to the physical device:
  looking at the device chain at '/sys/devices/pci0000:00/0000:00:1d.2/usb4/4-1/4-1:1.0':
    BUS=="usb"
    ID=="4-1:1.0"
    DRIVER=="usb_lpm"
    SYSFS{bAlternateSetting}==" 0"
    SYSFS{bInterfaceClass}=="07"
    SYSFS{bInterfaceNumber}=="00"
    SYSFS{bInterfaceProtocol}=="02"
    SYSFS{bInterfaceSubClass}=="01"
    SYSFS{bNumEndpoints}=="02"
    SYSFS{modalias}=="usb:v03F0p3404d0100dc00dsc00dp00ic07isc01ip02"

  looking at the device chain at '/sys/devices/pci0000:00/0000:00:1d.2/usb4/4-1':
    BUS=="usb"
    ID=="4-1"
    DRIVER=="usb"
    SYSFS{bConfigurationValue}=="1"
    SYSFS{bDeviceClass}=="00"
    SYSFS{bDeviceProtocol}=="00"
    SYSFS{bDeviceSubClass}=="00"
    SYSFS{bMaxPacketSize0}=="8"
    SYSFS{bMaxPower}==" 2mA"
    SYSFS{bNumConfigurations}=="1"
    SYSFS{bNumInterfaces}==" 1"
    SYSFS{bcdDevice}=="0100"
    SYSFS{bmAttributes}=="c0"
    SYSFS{configuration}==" "
    SYSFS{devnum}=="2"
    SYSFS{idProduct}=="3404"
    SYSFS{idVendor}=="03f0"
```

```

SYSFS{manufacturer}=="hp"
SYSFS{maxchild}=="0"
SYSFS{product}=="deskjet 6122"
SYSFS{serial}=="MY29K2B46033"
SYSFS{speed}=="12"
SYSFS{version}==" 2.00"

```

looking at the device chain at 'e/sys/devices/pci0000:00/0000:00:1d.2/usb4':

```

BUS=="usb"
ID=="usb4"
DRIVER=="usb"
SYSFS{bConfigurationValue}=="1"
SYSFS{bDeviceClass}=="09"
SYSFS{bDeviceProtocol}=="00"
SYSFS{bDeviceSubClass}=="00"
SYSFS{bMaxPacketSize0}=="8"
SYSFS{bMaxPower}==" 0mA"
SYSFS{bNumConfigurations}=="1"
SYSFS{bNumInterfaces}==" 1"
SYSFS{bcdDevice}=="0206"
SYSFS{bmAttributes}=="c0"
SYSFS{configuration}==" "
SYSFS{devnum}=="1"
SYSFS{idProduct}=="0000"
SYSFS{idVendor}=="0000"
SYSFS{manufacturer}=="Linux 2.6.14.3 uhci_hcd"
SYSFS{maxchild}=="2"
SYSFS{product}=="UHCI Host Controller"
SYSFS{serial}=="0000:00:1d.2"
SYSFS{speed}=="12"
SYSFS{version}==" 1.10"

```

looking at the device chain at '/sys/devices/pci0000:00/0000:00:1d.2':

```

BUS=="pci"
ID=="0000:00:1d.2"
DRIVER=="uhci_hcd"
SYSFS{class}=="0x0c0300"
SYSFS{device}=="0x265a"
SYSFS{irq}=="18"
SYSFS{local_cpus}=="1"
SYSFS{modalias}=="pci:v00008086d0000265Asv00001458sd0000265Abc0Csc03i00"
SYSFS{subsystem_device}=="0x265a"
SYSFS{subsystem_vendor}=="0x1458"
SYSFS{vendor}=="0x8086"

```

looking at the device chain at '/sys/devices/pci0000:00':

```

BUS==" "
ID=="pci0000:00"
DRIVER=="unknown"

```

È sufficiente identificare il campo maggiormente univoco. Ad esempio, a proposito dell'esempio sopra riportato delle due stampanti, se sono di una stessa marca è chiaro che il campo `SYSFS{manufacturer}` non sarà discriminante, mentre lo sarà `SYSFS{serial}`. Nel caso si abbiano due mouse identici, si ricorrerà al campo `{ID}` e via dicendo.

8.2.6 Creare regole per udev

Ma a che cosa può servire tutto ciò? Innanzitutto è possibile creare il node per un dispositivo come il microcodice della CPU, collocandolo in una diversa directory. La regola apparirà come segue:

```
KERNEL=="microcode", NAME="cpu/microcode"
```

Se in effetti il nome nel kernel è `microcode`, allora verrà creato il dispositivo `cpu/microcode`.

Allo stesso modo, si può far sì che la stampante non venga chiamate semplicemente `lp0`, ma col un nome indicativo, come `HP_DeskJet_6122`.

Se vogliamo creare dei node per per più dispositivi omologhi, la regola apparirà come segue:

```
KERNEL=="controlC[0-9]*", NAME="snd/%k"
```

In questo modo, per tutti i dispositivi il cui nome inizia con la stringa `controlC` (parliamo qui di espressioni regolari) verranno creati dei node nella directory `/snd` che prenderanno il nome assegnato loro dal kernel. `%k` è uno dei cosiddetti «operatori di sostituzione» di cui si vale la sintassi di udev.

Gli operatori di sostituzione sono:

%n il numero assegnato dal kernel al dispositivo.

%k il nome assegnato dal kernel al dispositivo.

%M il numero primario assegnato dal kernel al dispositivo.

%m il numero secondario assegnato dal kernel al dispositivo.

%b l'id nel bus del dispositivo.

%c la stringa prodotta dal programma esterno, specificata in PROGRAM.

%sfilename un attributo di sysfs.

%e se già esiste il node per un dispositivo, il più piccolo intero positivo decimale N è sostituito in modo che il nome risultante non coincida con nessun node già esistente (altrimenti non avrebbe luogo alcuna sostituzione). Ciò serve per creare compatibilità fra più `sysmlink` e per enumerare dispositivi dello stesso tipo, originati da differenti sottosistemi del kernel.

%% il carattere % stesso.

8.2.7 Alcuni esempi di regole per udev

L'utilità di simili procedure appare evidente se pensiamo a dispositivi (USB) rimovibili come un pendrive o una macchina fotografica digitale. Se infatti inseriamo in sequenza i due dispositivi, ma non sempre nello stesso ordine, avremo il caso in cui `sda1` sarà il pendrive, e quello in cui invece `sda1` indicherà la memoria di massa della macchina fotografica. Mediante le regole di udev è possibile associare ai dispositivi un vero e proprio nome distintivo fisso, che ci liberi da questa oscillazione.

Macchina fotografica digitale

Ipotizziamo di avere una macchina fotografica digitale. Con la regola:

```
BUS=="SCSI", SYSFS{vendor}=="FUJIFILM", SYSF{MODEL}=="M100", NAME=="camera"
```

facciamo in modo che, quando il dispositivo è inserito nella presa USB e rilevato da hotplug, udev gli assegni stabilmente il nome `camera`, dal momento che ne vengono prima rilevati il bus e il nome della marca.

Lettore CD e masterizzatore

Se vi vuole applicare una regola per definire stabilmente il lettore CD e il masterizzatore (ipotizzando che il primo si riferisca a `/dev/hdc` e il secondo a `/dev/hdd`), le due regole da creare (eliminando quelle preesistenti) saranno:

```
BUS=="ide", KERNEL=="hdc", NAME=="hdc", SYMLINK=="cd-rom, cdroms/cdrom%n" GROUP="cdrom"
BUS=="ide", KERNEL=="hdd", NAME=="hdd", SYMLINK=="cdrw cdroms/cdrom%n" GROUP="cdrom"
```

Poiché al riavvio di udev si avrà che `/dev/cdrom` punta a `/dev/hdc`, che `/dev/cdrw` punta a `/dev/hdd`, e che le due nuove regole di udev definiscono categoricamente i due link, in `/etc/fstab` l'avvio delle righe relative ai due dispositivi dovrà risultare come segue:

```
/dev/cdrom /mnt/cdrom ...
/dev/cdrw /mnt/cdrw ...
```

In questo modo, eventuali ulteriori modifiche ai dispositivi richiederanno soltanto la modifica del file con le regole e non anche di `/etc/fstab`, i cui link vengono risistemati automaticamente grazie a udev.

Due stampanti

Allo stesso modo è possibile risolvere il problema della eventuale presenza di due stampanti USB, magari una laser in bianco e nero (chiamata `lp_bw`) e una a getto d'inchiostro, a colori (chiamata `lp_color`). Il node del dispositivo sarebbe diverso se le inserissimo in momenti diversi nella presa USB, comportando problemi per la stampa e, ancor di più, per un'eventuale condivisione stampanti. Con una regola di udev possiamo invece fare in modo che a ogni stampante sia attribuito un identificativo univoco, del tutto indipendentemente dall'ordine con il quale esse vengono collegate al computer. La doppia regola assumerà dunque questo aspetto:

```
BUS=="usb", SYSFS{serial}=="W0909207101241330", NAME=="lp_color"
BUS=="usb", SYSFS{serial}=="HX0LL002202323480", NAME=="lp_b&w"
```

Quando udev rileva il primo numero seriale, lo associa al node dal nome `lp_color` e quando invece rileva il secondo, al node dal nome `lp_b&w`.

L'invocazione a programmi esterni

Un'altra importante funzione di udev è la possibilità di richiamare dei programmi esterni (come degli script) per determinare quale nome deve coincidere con un dispositivo, o se deve coincidere con esso. Ciò serve se le regole non sono sufficientemente elastiche. Ad esempio, se si vuole utilizzare un dato, diciamo il nome dell'artista, di un CD-ROM musicale, non c'è modo per poterlo fare mediante `sysfs`, che non ha nulla a che vedere coi contenuti del CD. Dobbiamo allora far sì che udev richiami il programma `name_cdrom.pl` (un ipotetico script Perl), non appena il kernel percepisce un dispositivo a blocchi. La regola assume la forma seguente:

```
KERNEL=="[hs]d[a-z]", PROGRAM=="name_cdrom.pl %M %m", NAME=="%c{1}" SYMLINK=="cdrom"
```

In questo modo la regola cerca ogni dispositivo che inizia con «h» o con «d» seguito da una «s» e quindi da una lettera qualsiasi dell'alfabeto. Se coincide con un qualche dispositivo, viene invocato lo script Perl `name_cdrom.pl`. In questo script sono forniti due argomenti, il minor e il major number del dispositivo (cioè `%M` e `%m`). Se il programma riesce, il primo esito fornisce il nome del dispositivo (cioè `%c1`). Viene così creato un link simbolico al file `/dev/cdrom`, cosicché gli altri programmi possono trovare il dispositivo `/dev/cdrom` stesso nel sistema. Lo script fa cercare il nome dell'artista nel database `freedb`, ammesso che il sistema sia connesso a Internet.

Scheda n. 16

I link simbolici

In GNU/Linux è possibile avere dei collegamenti a file o directory in maniera trasparente. Un link simbolico (detto anche *symlink* o *soft link*) è un file che punta a un altro file. Se per esempio volessimo (o avessimo la necessità di) chiamare lo stesso file con due nomi distinti, per evitare di ricopiare a ogni modifica uno dei due file (cosa che occuperebbe anche più spazio sul disco fisso), potremmo creare un link simbolico.

La sintassi del comando è: `ln -s /percorso/nome_file /percorso/nome_link`.

I link simbolici sono distinti dai file cui puntano, cioè all'inode number del primo ci sono le informazioni sull'inode number del secondo.

I link simbolici si differenziano dai cosiddetti *hard link*: questi ultimi sono riferimenti multipli che puntano infatti allo stesso inode number, quindi si hanno due repliche dello stesso file con lo spazio di uno. In tal modo un singolo file può essere raggiunto attraverso percorsi differenti. Essi si creano col comando: `ln /percorso/nome_file /percorso/nome_link`.

Di conseguenza, se si cancella il file puntato da un link simbolico, resterà un link «penzolante» (*dangling*) perché non esiste più il file a cui dovrebbe riferirsi. Se invece si cancella un file puntato da un *hard link*, non si farà altro che cancellare uno dei nomi con cui veniva chiamato quel file; il file viene effettivamente cancellato solo quando il conteggio dei link è uguale a 0.

Le uniche limitazioni nella creazione di *hard link* stanno nel fatto che essi non si possono riferire a directory, e nemmeno a file che risiedono su dispositivi differenti.

8.3 Il montaggio dei dispositivi

8.3.1 Il file `/etc/fstab`

Il file `/etc/fstab`, letto all'avvio dal sistema, si occupa di indicare al sistema come montare le varie unità, senza dover lanciare ogni volta il comando `mount`, mentre per smontare manualmente, il comando è `umount /dev/nome_dispositivo` oppure `umount nome_punto_di_montaggio`. È inoltre necessario aver creato nella directory `/mnt` dei punti di montaggio (che poi, a loro volta, non sono altro che delle directory), con dei nomi a piacere. In realtà, è possibile montare qualsiasi unità col comando `mount`, ma al riavvio del sistema il montaggio si perde. Con la procedura qui indicata invece il risultato viene fissato e si ha a ogni riavvio del sistema.

Un'interessante alternativa è il montaggio e smontaggio automatico dei dispositivi mediante `automount` (vedi [Scheda n. 15 – Automount e autofs](#)). La soluzione più recente consiste nel ricorso a `hal` e `dbus`, da cui tuttavia Slackware sembra ancora notevolmente distante.

8.3.2 Montaggio dei dispositivi presenti in `/etc/fstab`

Per montare tutti i dispositivi elencati nel file `/etc/fstab` si deve lanciare il comando: `mount -a`, mentre `mount` senza opzioni si limita a visualizzare i dispositivi attualmente montati. Per montare manualmente un dispositivo elencato in `/etc/fstab`, ma non caricato automaticamente, basta lanciare il comando: `mount` seguito dal punto di montaggio (ad esempio: `mount /mnt/floppy`) e per smontarlo il comando: `umount` seguito sempre dal punto di montaggio (ad esempio: `umount /mnt/floppy`).

8.3.3 Montaggio dei dispositivi non presenti in `/etc/fstab`

Se invece si deve montare un dispositivo non elencato in `/etc/fstab`, è necessario usare il comando `mount` con alcuni argomenti aggiuntivi. Sarà infatti necessario specificare almeno:

- a) il dispositivo
- b) il suo file system preceduto dall'opzione `-t`
- c) il suo punto di montaggio.

Ad esempio, per montare un floppy formattato FAT e collocato nella prima unità floppy: `mount /dev/fd0 -t fat /mnt/floppy`; quindi, per smontarlo, ancora una volta `umount /mnt/floppy`. Un eventuale messaggio del tipo «device busy» significa che il dispositivo è in uso, cioè che qualche applicazione vi sta accedendo; sarà necessario chiuderla per poter effettuare lo smontaggio.

Scheda n. 17

Il file `/etc/fstab`

Ogni riga del file è dedicata a un determinato dispositivo, secondo una sintassi del tipo:
[dispositivo] [punto di montaggio] [file system] [opzioni di montaggio] [dump]
[check]

I dispositivi

Ogni tipo di dispositivo è indicato da una sigla. Vediamo i principali.

hd disco fisso IDE. In particolare si avrà `hd xn` , dove x indica il disco e n la partizione. Come sappiamo, `hda` è il master sul controller primario e `hdb` lo slave sul controller primario; `hdc` è il master sul controller secondario e `hdd` lo slave sul controller secondario. Coi numeri invece vengono indicate le partizioni. Dunque `/dev/hda1` indicherà la prima partizione del primo disco rigido, `/dev/hda2` la

seconda, `/dev/hdb1` la prima partizione del secondo disco rigido e `/dev/hdb2` la seconda partizione del secondo disco rigido ecc. In linea di massima, anche i lettori di CD e i masterizzatori IDE vengono identificati con questo tipo di dispositivo. Inoltre Slackware crea di default `/cdrom`, un collegamento simbolico al dispositivo CD-ROM (generalmente `/dev/hdc` o `/dev/hdd`): infatti `/dev/cdrom` non è che un link simbolico a `/dev/hdc` o a `/dev/hdd`.

sd disco SCSI. Come per i dischi IDE, si avrà una successiva lettera e dei numeri per indicarne la partizione. Di conseguenza `/dev/sda` sarà primo dispositivo scsi e `/dev/sdb` il secondo. In linea di massima, anche i lettori di CD e i masterizzatori SCSI vengono identificati con questo tipo di dispositivo, nonché alcune unità rimovibili come i pendrive, le memorie di massa delle macchine fotografiche digitali, i dischi USB esterni. In questo caso, eventuali partizioni saranno indicate nel modo consueto (`sda1`, `sda2` ecc.).

fd il dispositivo floppy. Il primo dispositivo floppy è indicato con `/dev/fd0`; nel caso (ormai raro) ve ne siano più d'uno, essi vengono numerati progressivamente.

Il punto di montaggio

Come abbiamo visto, il sistema deve montare un dispositivo in una determinata cartella; la soluzione migliore (se non ci sono esigenze particolari) è crearla nella directory `/mnt`, che il sistema stesso ci propone come cartella di default per questo tipo di operazioni (anche se è possibile creare altre cartelle a piacere).

I file system

Come abbiamo visto, GNU/Linux supporta una enorme quantità di file system. Abbiamo già menzionato i più diffusi. In `/etc/fstab` è però possibile impostare questa voce ad auto perché il sistema cerchi di determinare automaticamente di quale tipo di file system si tratta. Ciò è utile nel caso di floppy o di pendrive, che possono essere formattati di volta in volta in diverso modo, e ci esime dal modificare le voci del file.

Le opzioni di montaggio

Le più comuni e utili (per un elenco completo, si veda `man mount`) sono:

auto monta automaticamente il dispositivo all'avvio.

noauto non monta automaticamente il dispositivo all'avvio. Il dispositivo dev'essere montato da shell col comando `mount`, oppure, in un server X, anche con l'apposita procedura (in KDE cliccando col mouse sull'icona del dispositivo presente sul desktop). È ovvio che unità non sempre presenti, come floppy o pendrive, andranno caratterizzate con questa opzione: in caso contrario il sistema cercherà comunque di caricarle e, non trovandole, ci darà un messaggio di errore.

defaults monta il dispositivo secondo i permessi di default (che sono definiti in `/etc/profile`).

user consente a un utente non root di montare e smontare il dispositivo.

users consente a tutti gli utenti non root di montare e smontare il dispositivo (vale a dire che un dispositivo montato da un certo utente può essere smontato da un altro, cosa impossibile con l'opzione «user»).

nousers consente solo all'utente root di montare il dispositivo.

owner consente solo al proprietario di montare il dispositivo.

ro consente il montaggio del dispositivo in sola lettura («read only»).

rw consente il montaggio del dispositivo in lettura e scrittura («read & write»).

uid stabilisce il proprietario del dispositivo.

gid stabilisce quale gruppo possa accedere al dispositivo.

umask stabilisce i parametri di accesso al dispositivo per utenti e gruppi.

noatime non aggiorna l'ora di accesso all'inode per ogni accesso, velocizzandolo.

exec consente l'esecuzione di file binari (il contrario è **noexec**).

sync sincronizza le operazioni di scrittura sui dispositivi. Può risultare utile nel caso di dispositivi rimovibili come floppy e pendrive perché, pur rallentando tali operazioni, è una garanzia contro la perdita di dati.

L'opzione dump

Questo valore è usato dal comando `dump` per effettuare dei backup sui file system. Il valore deve essere impostato a 0 per evitare l'effettuazione del backup, a 1 per consentirla. Di norma questa opzione si imposta a 1 solo per i file system non rimovibili.

L'opzione check

Questo valore è usato dal comando `check` per determinare l'ordine del controllo dell'integrità del file system all'avvio del sistema. Può assumere i valori:

0 = nessun controllo

1 = controlla per primo

2 = controlla in seguito

Generalmente i valori sono:

1 per la partizione di root (/)

2 per le altre partizioni dei dischi rigidi

0 per tutti gli altri dispositivi.

I file system presenti nello stesso disco sono verificati in sequenza, mentre è possibile abilitare il check contemporaneo di file system collocati in dischi fissi differenti. Di conseguenza è sconsigliato assegnare il valore 1 a partizioni diverse dello stesso disco fisso, perché il lavoro delle testine non risulterebbe ottimizzato.

La verifica dei file system

Per modificare la frequenza dei controlli su file system EXT2 ed EXT3 si ricorre a `tune2fs`, secondo la sintassi `tune2fs [opzioni] [argomenti] [file system]`. Di default la verifica viene compiuta ogni centottanta giorni od ogni venti riavvii o montaggi. Con l'opzione `-c` esso imposta il numero massimo di montaggi tra due verifiche del file system: così, il comando: `tune2fs -c 30 /dev/hda1` fa sì che, se non intervengono necessità particolari, il controllo di `/dev/hda1` sarà effettuato ogni trenta montaggi e non di più. L'opzione `-i` imposta l'intervallo temporale massimo tra le verifiche, e assume tre argomenti: `d`, `m` e `w`. Il postfisso `d` (oppure nessun postfisso) forza una verifica giornaliera, il postfisso `w` una settimanale e il postfisso `m` una mensile. Il valore 0 (zero) disabilita la dipendenza temporale delle verifiche. Infine, l'opzione `-e` imposta il comportamento che il kernel deve assumere se vengono rinvenuti errori nel file system, e assume tre valori: `continue`, cioè continua la normale esecuzione; `remount-ro`, cioè rimonta il file system in sola lettura (read-only); `panic`, cioè provoca un kernel panic bloccando il sistema. Le opzioni possono anche essere combinate; per disabilitare ogni controllo su `/dev/hda1` il comando sarà: `tune2fs -c 0 -i 0 /dev/hda1`.

Questa utilità non dev'essere mai usata su file system montati in lettura e scrittura, i quali andranno perciò precauzionalmente montati in sola lettura.

La verifica forzata del file system si ottiene, per EXT2 ed EXT3, con `e2fsck`, con la sintassi: `e2fsck [opzioni] [file system]`. Se usato su EXT3, `f2fsck` applica il journal prima di procedere con la verifica. Le opzioni principali sono: `-b` che cerca un superblocco diverso da quello normale, `-c`, che avvia il programma `badblocks` per trovare e quindi marcare i settori danneggiati del file system, `-f`, che forza il controllo, `-n` che apre il file system in sola lettura e assume la risposta «no» a tutte le domande (ed è il contrario dell'opzione `-y`), consentendo di usare `f2fsck` in modo non interattivo (è una modalità di sicurezza), `-p` (= «preen») che ripara automaticamente il file system.

Questa utilità non dovrebbe mai essere usata su file system montati, le quali andranno perciò precauzionalmente smontate.

Se il file system sembra presentare degli errori, si può scaricare il journal nel database del file system col comando: `e2fsck -fy /dev/periferica` (dove *periferica* sarà indicata con la lettera dell'unità e il numero della partizione) e quindi procedere a un nuovo controllo. Se il superblocco normale risulta danneggiato, si possono ottenere i numeri dei superblocchi di backup col comando: `mke2fs -e periferica` e quindi sostituire il superblocco danneggiato con quello di backup, col comando `fsck -b num`, dove *num* è il numero del superblocco di backup.

Per modificare le impostazioni dei controlli su file system Reiserfs si ricorre a `reiserfstune`, secondo la sintassi `reiserfstune [opzioni] [file system]`. Questa utilità è in grado di modificare due parametri del journaling, ossia la sua dimensione e quella della transazione massima, in blocchi. L'opzione più utile è `-f`, che forza l'aggiornamento del journal anche per i file system creati prima del codice di rilocalizzazione del journal. Il journal può risiedere anche in file system diversi da quello attuale.

La verifica forzata del file system si ottiene, per Reiserfs, con `reiserfsck`, con la sintassi: `reiserfsck [opzioni] [file system]`. Fra le opzioni disponibili, `--check` avvia la verifica della consistenza del file system senza però ripararlo in caso di errori (da usarsi in file system montati in sola scrittura); `--fix-fixable` ricostruisce la consistenza laddove non ci sia bisogno di ricostruire l'intero albero del file system (è lo stesso `reiserfs --check` ad avvertire se questa opzione è opportuna); `--rebuild-tree` ricostruisce l'intero file system utilizzando i «nodi foglia» (*leaf nodes*) trovati nel dispositivo (è lo stesso `reiserfs --check` ad avvertire se questa opzione è necessaria); `--rebuild-sb`, che ricostruisce il superblocco se il sistema sembra incapace di trovare una partizione formattata Reiserfs mentre noi siamo sicuri che c'è.

In aggiunta, l'opzione `-S` scansiona l'intera partizione e non solo lo spazio utilizzato.

Alla fine di `/etc/fstab` è necessario lasciare sempre una riga bianca in più, o riceveremo un messaggio di errore del tipo «warning: no final newline at the end of `/etc/fstab`».

Un esempio

Ecco un esempio di `/etc/fstab`, che include anche configurazioni particolari che verranno trattate nel prosieguo (la condivisione NFS):

```
/dev/hda4      swap          swap          defaults      0 0
/dev/hda2      /             ext3          defaults      1 1
/dev/hda3      /home        ext3          defaults      1 2
/dev/hdb1      /mnt/archivio ext3          defaults      1 2
/dev/hdb3      /mnt/slack   ext3          defaults      1 2
/dev/hdb4      /mnt/debian  ext3          defaults      1 2
/dev/hda1      /mnt/windows ntfs         auto,umask=022,ro 0 0
/dev/hdb2      /mnt/backup  vfat          defaults      0 0
/dev/cdrom     /mnt/cdrom   iso9660       noauto,users,ro 0 0
/dev/dvd       /mnt/dvd     iso9660       noauto,users,ro 0 0
/dev/fd0       /mnt/floppy  auto          noauto,users  0 0
/dev/sda1      /mnt/pendrive vfat          noauto,users  0 0
192.168.0.3:/  mnt/brightstar nfs           noauto,_netdev,sync,users 0 0
```

```
devpts      /dev/pts      devpts      gid=5,mode=620      0 0
proc        /proc         proc        defaults             0 0
```

8.4 Montaggio dei dischi rigidi

Per montare partizioni del disco rigido o un altro disco rigido, è necessario editare `/etc/fstab` aggiungendo ad esempio, a seconda dei casi, righe come le seguenti:

```
/dev/hda1  /mnt/windows  ntfs  auto,defaults,umask=022,ro  0 0
/dev/hda5  /mnt/archivio  vfat  auto,gid=104,umask=004     1 0
```

posto che si sia già provveduto a creare in `/mnt` i rispettivi punti di montaggio, e cioè `/mnt/windows` e `/mnt/archivio` (i nomi delle directory in `/mnt` sono scelti a piacere).

Nel primo caso abbiamo indicato: a) la partizione del disco fisso che contiene il programma; b) la directory di montaggio; c) il tipo di file system; d) abbiamo dato indicazione di montarla automaticamente (cioè ogni volta all'avvio), e) abbiamo abilitato il montaggio di default, f) abbiamo tuttavia modificato i permessi con `umask` in modo che gruppo e utenti possano accedere all'unità, g) abbiamo specificato che si tratta di una unità in sola lettura; f) abbiamo infine escluso il backup e il controllo del file system.

Nel secondo caso con `gid` e `umask` combinati abbiamo autorizzato un ipotetico gruppo 104 (il gruppo «archivio») a effettuare tutte le operazioni, e gli altri utenti ad accedere in sola lettura.

I permessi per il montaggio delle unità non riguardano naturalmente la possibilità di accedervi da parte degli utenti. In una unità disco di proprietà di `root:root` si possono creare directory appartenenti a utenti diversi, che avranno perciò la piena possibilità di utilizzarle.

8.5 Montaggio di lettori CD e masterizzatori o unità DVD

Per montare un lettore CD, un masterizzatore o una unità DVD, è necessario creare un punto di montaggio in `/mnt`, sempre con un nome a piacere, qui ad esempio `/mnt/cdrom` per il lettore CD e `/mnt/dvd` per l'unità DVD. Quindi, se sulla seconda porta (IDE2) il masterizzatore è master e il lettore è slave, bisogna editare `/etc/fstab` aggiungendo le seguenti righe (altrimenti basta invertirle):

```
/dev/hdc   /mnt/dvd      iso9660,udf  noauto,users,ro  0 0
/dev/hdd   /mnt/cdrom    iso9660,udf  noauto,users,ro  0 0
```

In questo caso il file system è `iso9660`, il formato standard per i CD, oppure `udf`. L'opzione `users` è utile per consentire a tutti gli utenti di montare e smontare queste unità. L'opzione `defaults` invece consentirebbe il montaggio e lo smontaggio solo a `root`, visto che questi dispositivi appartengono a `root:root`.

Un'ultima cautela per usare il masterizzatore: col kernel 2.4 e con le vecchie versioni di `cdrecord` (il programma che serve a scrivere i CD) è necessario abilitare l'emulazione SCSI. A questo fine è sufficiente aggiungere alla fine di `/etc/lilo.conf`, con un editor qualsiasi di testo, la riga:

```
append='hdc=ide-scsi hdd=ide-scsi'
```

In questo caso, lettore e masterizzatore non saranno più visti come dispositivi IDE, ma appunto come SCSI, per cui non saranno indicati più – poniamo – con `/dev/hdc` e `/dev/hdd`, bensì come `/dev/scd0` e `/dev/scd1`. In ogni caso, non è inopportuno abilitare l'emulazione SCSI anche con le versioni più recenti di `cdrecord`, il che velocizza notevolmente (con una certa approssimazione, di un 40%) la scrittura.

Andrà quindi controllato il link simbolico `/dev/cdrom` (creato durante l'installazione), ed eventualmente cancellato e ricreato per fare in modo che corrisponda al nuovo dispositivo. I passaggi sono semplicemente: `rm /dev/cdrom` e `ln -s /dev/sdc0 /dev/cdrom` nel primo caso e, nel secondo, `rm /dev/dvd` e `ln -s /dev/sdc1 /dev/dvd`.

Ipotizziamo infine di avere un disco fisso SATA, identificato con `/dev/sda` e un lettore CD-ROM IDE identificato di default con `/dev/hda`. Attivando l'emulazione SCSI tale lettore verrà ora identificato con un dispositivo SCSI, in genere `/dev/sr0` (o `/dev/sg0`), reperibile mediante `dmesg | grep sr0` (o `dmesg | grep sg0`). Di conseguenza anche in questo caso il link andrà ricreato con la procedura sopra indicata.

Col kernel 2.6 è sempre possibile abilitare l'emulazione, ma le versioni più recenti di `cdrecord` non lo richiedono più (e anzi lo dichiarano deprecato), lavorando tranquillamente e con maggior efficienza coi masterizzatori IDE, per cui non serve aggiungere nulla in `/etc/lilo.conf`.

8.6 Montaggio di un pendrive USB

Per montare un pendrive USB, è necessario creare un punto di montaggio in `/mnt`, con un nome a piacere, ad esempio `/mnt/pendrive`. Quindi editare `/etc/fstab` aggiungendo la seguente riga:

```
/dev/sda1 /mnt/pendrive vfat noauto,users,icharset=iso8859-15 0 0
```

(ipotizziamo qui che il file system utilizzato sia `vfat` e che la codifica del linguaggio sia l'italiano; quest'ultima indicazione è del tutto opzionale. Se il pendrive fosse formattato FAT, i nomi dei file risulterebbero di soli otto caratteri, più l'estensione).

Se poi compileremo il kernel 2.6, non tralasciamo di abilitare il supporto per i dischi SCSI e per l'hot-plug, dal momento che questi dispositivi (e altri supporti come i dischi Zip) vengono riconosciuti per l'appunto come dischi SCSI. In caso di problemi, è innanzitutto opportuno verificare con che nome (device) viene riconosciuto il dispositivo. Lo si può fare inserendo il pendrive nella presa USB e controllando immediatamente l'output del comando: `tail -F /var/log/messages`, oppure lanciando il comando: `sfdisk -l /dev/sda` (oppure `sdb` ecc. se il disco fisso è SATA e dunque `sda` indica già un'altra unità). È anche possibile, ma in genere superfluo, abilitare gli utenti al montaggio aggiungendone i nomi nel file `/etc/group` alla voce `disk`.

Scheda n. 18

Automount e autofs

Il servizio di automount

L'automount è un processo che consente ad alcuni file system (come i dispositivi rimovibili e le condivisioni NFS) di essere montati in maniera automatica, quando un utente cerca di accedervi. Allo stesso modo, il file system viene smontato dopo un periodo (convenzionalmente stabilito) in cui il file system non viene più usato.

Slackware, pur non avendo già attivo un sistema di montaggio e smontaggio automatico dei dispositivi rimovibili (CD-ROM, pendrive, dischi USB) e delle condivisioni NFS, è predisposta per questo tipo di servizio. Dei file di configurazione di esempio sono presenti in `/usr/doc/autofs-3.1.7/samples`.

Questo servizio, che naturalmente si può attivare anche nei computer desktop, risulta però di particolare utilità in ambienti di rete molto estesi. Poiché infatti i dispositivi montati da `/etc/fstab` occupano comunque delle risorse, l'automount consente di montarli solo quando se ne ha effettiva necessità e di smontarli automaticamente quando, per un certo periodo, non se ne fa uso. È altresì utile in presenza di molti mount incrociati anche fra pochi computer, nel caso ve ne siano di inattivi (e infine in casi particolari, ad esempio quando sia presente la necessità di operare il montaggio di unità DOS in modalità di conversione attivata e disattivata insieme).

Questo servizio è affidato ad **autofs** e ad **automount**. Autofs serve a rilevare le richieste di montaggio, e quando viene effettuata una richiesta invoca automount per compiere effettivamente l'operazione di montaggio.

Si presuppone che sia attivato nel kernel (nella sezione «File systems») il supporto per autofs4, che rimpiazza il vecchio autofs, dichiarato obsoleto. Mentre autofs era implementato in user-space e si basava sul demone AMD che passava il traffico ai file system automontati attraverso NFS, autofs4 (dal kernel 2.2) è implementato a livello kernel. Ciò significa che, in questo caso, è il kernel a conoscere quali sono i punti di montaggio dei sistemi automontati. Se nel kernel è attivato solo il supporto ad autofs4 (e non anche quello ad autofs), per ragioni di compatibilità potrebbe essere necessario inserire in `/etc/modprobe.conf` la seguente linea:

```
alias autofs autofs4
```

L'inizializzazione del servizio

Per inizializzare il servizio, si può inserire in `/etc/rc.d` uno script chiamato `rc.autofs`, da rendere ovviamente eseguibile. Inoltre bisogna inserire un'invocazione a questo script. Le possibilità a questo fine sono due: o si aggiunge a `/etc/rc.d/rc.local` la seguente riga:

```
/etc/rc.d/rc.autofs start
```

oppure si aggiunge a `/etc/rc.d/rc.M` una sezione come la seguente:

```
# Start the automount demon:
if [ -x /etc/rc.d/rc.autofs ]; then
    . /etc/rc.d/rc.autofs start
fi
```

Come già detto, un esempio generico è contenuto in `/usr/doc/autofs-3.1.7/samples`; esso si chiama `rc.autofs.in`, va rinominato `rc.autofs` e reso eseguibile con `chmod`. Tuttavia, a meno di non modificarlo in misura abbastanza cospicua, non funziona perché è concepito per distribuzioni RedHat o Debian. Alcuni script adatti specificamente a Slackware (fermo restando il ricorso ai motori di ricerca) si possono reperire rispettivamente agli indirizzi:

<http://www.slackware.com/~david/zuul/nis-mini-howto/rc.autofs> (Cantrell)

<http://mariner.cs.ucdavis.edu/slackware/doc/slackware/rc.autofs> (Terminator)

<http://www.vivaolinux.com.br/artigos/impressora.php?codigo=734> (Ribeiro)

<http://schelcj.avlug.org/files/rc.autofs.patch> (Scheller)

In alcuni di questi script è già incluso il timeout, che non andrà pertanto specificato in `/etc/auto.master`. Ad esempio, nello script di Ribeiro si legge una sezione dedicata per appunto a stabilire il timeout.

I comandi per gestire manualmente autofs sono:

```
/etc/rc.d/rc.autofs start, per avviare il processo;
```

```
/etc/rc.d/rc.autofs stop, per terminare il processo;
```

```
/etc/rc.d/rc.autofs status, per vedere la configurazione e i sistemi attualmente automontati;
```

`/etc/rc.d/rc.autofs reload`, per riavviare il demone dopo eventuali modifiche.

Se invece si vuole far partire il servizio senza ricorrere allo script `rc.autofs`, basta inserire in `/etc/rc.d/rc.local` una riga del tipo:

```
/usr/sbin/automount --timeout 10 /misc file /etc/auto.misc
```

dove sono indicati il percorso dell'eseguibile, il tempo (in secondi, in questo caso 10) dopo cui un dispositivo montato e non più utilizzato viene automaticamente smontato, la directory di montaggio dei dispositivi (`/misc`), il tipo di configurazione adottato (in questo caso «file») e infine il file stesso di configurazione, che ha un nome convenzionale; in questo caso, seguendo il file di esempio di Slackware, `/etc/auto.misc`: solo la prima parte del nome del file, «auto» dev'essere sempre presente, la seconda parte è un nome a piacere (perciò `auto.mount`, `auto.dev`, `auto.misc` ecc.).

Il file `/etc/auto.master`

Il file `/etc/auto.master` configura l'`automount` indicando la directory radice di montaggio dei sistemi automontati, in questo caso `/misc`, il percorso del file di configurazione (detto «file di mappa») e il timeout. Ovvio che i dati devono corrispondere con quelli indicati in `/etc/rc.d/rc.local`. Un tipico `/etc/auto.master` si presenta così:

```
# $Id: auto.master,v 1.2 1997/10/06 21:52:03 hpa Exp $
# Sample auto.master file
# Format of this file:
# mountpoint map options
# For details of the format look at autofs(8).
/misc /etc/auto.misc --timeout 10
```

L'operazione svolta da `auto-master` si può effettuare anche da shell, col comando:

```
automount [opzioni] [punto_di_montaggio] [tipo di configurazione]
```

Opzione necessaria è il timeout, espresso in secondi. Il mount point è, come sappiamo, la directory in cui saranno montati i dispositivi (e che dev'essere vuota). Il tipo di configurazione nel nostro caso è un file. Infine si indica il file di mappa. Il comando sarà dunque del tipo:

```
automount -t 10 /misc file /etc/auto.misc
```

Il file `/etc/auto.misc`

Il file `/etc/auto.misc` è il file di mapping tra file system da montare e dispositivi, ossia indica i dispositivi che devono essere montati con l'`automount`, precisandone il file system e le opzioni di montaggio. Gli elementi da indicare sono i seguenti:

- nome del dispositivo e suo punto di montaggio (detto «chiave», che poi diventerà `/misc/chiave`),
- file system ed eventuali opzioni, separate da una virgola (senza spazi intermedi),
- provenienza del file system. Nel caso dei dispositivi rimovibili si indica il nome del dispositivo come appare in `/dev`, preceduto da un doppio punto.

Nel caso di condivisioni NFS si indica l'IP del computer da condividere (o il suo nome, se è attivato il servizio di risoluzione dei nomi), eventualmente preceduto dal nome degli utenti che hanno il diritto ad accedervi e seguito dalle directory effettivamente condivise (secondo un sistema analogo a quello usato per impostare la condivisione NFS). Si nota come, per lo più, le opzioni sono le medesime di `/etc/fstab`, ad eccezione di `-fstype`. Ecco un tipico esempio di `/etc/auto.misc`:

```
# $Id: auto.misc,v 1.2 1997/10/06 21:52:04 hpa Exp $
# This is an automounter map and it has the following format
# key [ -mount-options-separated-by-comma ] location
# Details may be found in the autofs(5) manpage

kernel          -ro                               ftp.kernel.org:/pub/linux
boot            -fstype=ext3                       :/dev/hda1
```



```
cd          -fstype=iso9660,ro      :/dev/hdc
cdrw        -fstype=iso9660,ro      :/dev/hdd
floppy      -fstype=auto,umask=000  :/dev/fd0
pendrive    -fstype=vfat,umask=000  :/dev/sda1
brighstar   -fstype=nfs,soft,intr   192.168.0.3:/
```

Qui si montano un lettore CD (**cd**), un masterizzatore (**cdrw**), il floppy (**floppy**), un pendrive (**pendrive**), e una condivisione NFS (**brighstar**) Per completare la configurazione, i dispositivi da montare con automount vanno eliminati da `/etc/fstab` (è sufficiente commentare le righe relative).

Il montaggio dei dispositivi

La chiamata al dispositivo si può effettuare da shell col semplice comando: `cd` seguito dall'indirizzo e dal nome del dispositivo come presente in `/etc/auto.misc` (ad esempio `cd /misc/floppy`), oppure, in KDE, creando sul desktop un'icona per il dispositivo a cui attribuire un URL che sarà il punto di montaggio (allo stesso modo, `/misc/floppy`). Per poter estrarre il dispositivo rimovibile, basta attendere il tempo di timeout, se nessuna applicazione lo sta utilizzando.

Capitolo 9

L'installazione delle periferiche

9.1 Installare una stampante

9.1.1 Avviare Cups

Se abbiamo settato Slackware in modo che all'avvio carichi automaticamente CUPS, e cioè il server di stampa, dobbiamo ora impostarlo. In caso contrario, è semplicissimo determinarne l'avvio ricorrendo a **pkgtool**. Si lancia `pkgtool`, si sceglie la voce «setup», si seleziona la voce «services» con la barra spaziatrice e si dà **Invio**, infine si seleziona «rc.cups». In questo modo viene reso eseguibile `/etc/rc.d/rc.cups` (il che si può ottenere anche con `chmod`).

CUPS supporta un elevato numero di modelli di stampanti Apple, Canon, Dymo, Epson, Hewlett-Packard, Lexmark, Okidata e Zebra. Se il modello che possediamo non fosse tra questi, dovremo ricercare i driver nel sito <http://www.linuxprinting.org>. Possiamo verificare l'avvenuto riconoscimento della stampante col comando `lpstat -p`.

9.1.2 Installare una stampante USB

Se la stampante è collegata a una porta USB ed è supportata da CUPS, la configurazione può avere luogo immediatamente, via browser. Apriamo da un browser l'indirizzo IP `http://localhost:631`, a cui si accede solo come root, e si aprirà la finestra di configurazione di CUPS. Scegliamo la voce «Do Administration Task», quindi «Printers», infine «Add Printer» e seguiamo la procedura che ci viene indicata. È sufficiente inserire un nome a piacere per la stampante, la sua collocazione (se essa è connessa al nostro computer, possiamo scrivere «locale»), e una eventuale breve descrizione. Scegliamo il dispositivo, e cioè la porta, che sarà verosimilmente «USB Printer #*n*» (dove *n* è il numero della stampante): poiché la stampante dovrebbe essere già stata rilevata, dovremmo riscontrarne fra parentesi il nome. Scegliamo la marca, quindi il modello e il driver suggerito. Un avviso ci comunicherà che la stampante è stata correttamente installata. Cliccando sul nome, possiamo applicare gli ultimi ritocchi, ad esempio modificando il formato della carta da Letter (ben poco diffuso in Italia) ad A4 e scegliendo risoluzione e modalità di stampa.

Se la stampante non è un modello o una marca supportata da CUPS, bisogna comperare una ricerca in rete per recuperare il driver, ossia il file `.ppd` specifico (ormai solo pochi vecchi modelli non sono supportati da GNU/Linux e alcuni recenti modelli di stampanti multifunzione. Peraltro, in `/testing` Slackware 11.0 presenta il pacchetto **hplip**, che supporta non solo tutte le stampanti Hewlett Packard già supportate da **hpijs**, ma in aggiunta molti modelli multifunzione, per un numero che supera il migliaio). I driver disponibili su <http://www.linuxprinting.org> sono in formato Ghostscript, ma possono essere convertiti automaticamente, nel sito stesso, in file `.ppd`. Una volta recuperato il driver, es-

so va collocato nella cartella `/usr/share/cups/model` e dotato dei permessi corretti col comando: `chmod 644 /usr/share/cups/model/nome_stampante.ppd`. A questo punto, basta seguire la procedura sopra descritta.

9.1.3 Installare una stampante parallela

Analoga è la procedura per stampanti a porta parallela. Apparentemente in Slackware il supporto per stampante a porta parallela è disabilitato di default, almeno stando a `/etc/rc.d/rc.modules`; in realtà esso è attivo, come si può verificare col comando `dmesg | grep lp`; in caso contrario, lo si dovrà attivare o ricompilando il kernel e inserendovi il supporto, oppure abilitando il caricamento del modulo. In quest'ultimo caso, nel file `/etc/rc.d/rc.modules` andrà decommentata la sezione seguente:

```
#### Parallel printer support ####
if cat /proc/ksyms | grep "[lp]" 1> /dev/null 2> /dev/null ; then
    echo "lp support built-in, not loading module" > /dev/null
else
    if [ -r /lib/modules/$RELEASE/kernel/drivers/char/lp.o \
        -o -r /lib/modules/$RELEASE/kernel/drivers/char/lp.o.gz \
        -o -r /lib/modules/$RELEASE/kernel/drivers/char/lp.ko ]; then
        /sbin/modprobe lp
    fi
fi
```

Quindi va fatto partire o ripartire il demone `cupsd`, rispettivamente col comando `/etc/rc.d/rc.cups start` oppure `/etc/rc.d/rc.cups restart` e si può infine passare alla configurazione descritta sopra via browser. La stampante, che dovrebbe già essere stata rilevata, si installerà in questo caso in `/dev/lpn`.

Scheda 19

I server di stampa di GNU/Linux

In Unix esistono vari server di stampa (Berkeley LPD, LPRng, Ipsched, CUPS). GNU/Linux ha ereditato LPD e CUPS. Fino alla versione 9.1 Slackware proponeva come default **LPD**: esso è stabile, ma carente di funzioni e a volte complicato da configurare. Dalla versione 10.0 il default è invece **CUPS**. Esistono vari tipi di filtri: i due più importanti sono il più vecchio **ifhp**, filtro generico per le stampanti PostScript, e il più recente **Foomatic**, che funziona su tutti i server e gestisce Ghostscript per le stampanti che non comprendono PostScript. Esistono infine vari client: i due tradizionali sono **lpr** e **lp**. Mentre le applicazioni grafiche in genere convertono automaticamente i formati dei file (siano essi testi o immagini; ad esempio Gimp ha un proprio accessorio di conversione e cioè `gimp-print`) in PostScript, la stampa da riga di comando pone problemi assai maggiori, perché è necessario sapere esattamente come e quando effettuare la conversione prima di inviare l'output alla stampante.

Questi componenti servono per effettuare i vari passaggi che il processo di stampa richiede. In sintesi, un'applicazione (come un editor) genera un documento da stampare, il quale viene inviato da un client di stampa a un server di stampa e immesso in una coda di stampa, una sorta di «lista d'attesa». Quindi il server invia il documento a un programma detto filtro di stampa che lo modifica per renderlo accettabile per la stampante. Se la stampante supporta PostScript, il server invia il risultato di questa operazione direttamente alla periferica di stampa, altrimenti lo invia prima a un altro filtro che rasterizza il documento (conferendogli una forma bitmap), e quindi alla stampante.

Le applicazioni di stampa GNU/Linux (e in generale Unix) generano file PostScript, che non è tanto un formato di file, quanto un linguaggio di descrizione vettoriale della pagina, cioè un programma. Per questo esso necessita di un interprete che lo visualizzi. Questa conversione si chiama, per l'appunto, rasterizzazione, ed è effettuata dal programma **Ghostscript**.

Cups (acronimo di «Common UNIX Printing System») consente di configurare stampanti locali e remote sia col protocollo **IPP** (Internet Printing Protocol) sia coi protocolli **LPD** (Line Printer Daemon) e **SMB** (Server Message Block); questi ultimi due sono però supportati con funzionalità ridotte. Per riconoscere le caratteristiche delle stampanti CUPS usa dei file .ppd allocati in /usr/share/cups/model: si tratta di file che contengono istruzioni di tipo MIME (CUPS è quasi sempre in grado di determinare il tipo MIME e cioè il formato di ciascun documento e di farne operare la corretta conversione prima di inviarlo alla stampante). Per svolgere questa funzione CUPS si basa su due file di database: /etc/cups/mime.types e /etc/cups/mime.convs; il primo contiene un elenco dei formati di file che CUPS può gestire, mentre il secondo associa ciascun tipo MIME con il programma in grado di elaborarlo. I file .ppd sono di solo testo e descrivono le caratteristiche e le capacità di una determinata stampante.

Cups gestisce in proprio i filtri di stampa incorporando dei programmi specifici, ma si serve anche di programmi esterni, come **ghostscript** (un interprete per i formati PostScript e Adobe Portable Document), **hpijs** (che fornisce il supporto a un'ottantina di stampanti Hewlett-Packard) e **espgs** (che supporta il formato pdf). Attualmente, oltre a hpijs, è disponibile anche **hplip** (che contiene tutti i driver di hpijs, ma supporta in aggiunta le stampanti HP multifunzione), non presente però nei pacchetti ufficiali di Slackware. CUPS supporta un gran numero di stampanti e di opzioni, ed è semplice da configurare, perché offre, fra l'altro, una comoda interfaccia web.

Il principale file di configurazione di CUPS è /etc/cups/cupsd.conf. Esso presenta delle falle di sicurezza, perché invia in rete dati come la password di configurazione in forma di testo semplice. Modificare questo e altri aspetti è possibile, ma interessa più un amministratore di rete che un utente domestico. Le operazioni di configurazione di CUPS si possono effettuare dall'interfaccia web che si apre all'indirizzo <http://localhost:631>, oppure anche, naturalmente, da riga di comando.

9.1.4 Condividere una stampante tra sistemi GNU/Linux

Condividere la stampante significa permettere a dei computer remoti GNU/Linux di stampare con una stampante fisicamente connessa al server di stampa.

Partiamo dal presupposto che sul server la stampante sia già stata configurata mediante CUPS. Prendiamo nota dell'indirizzo IP della macchina (come 192.168.0.2) e del nome con cui abbiamo registrato la stampante (ad esempio hp_deskjet_6122). Dobbiamo apportare una modifica nel file /etc/cups/cupsd.conf con un editor di testo per aggiungere gli host autorizzati a usufruire della stampante fisicamente connessa al server. Troviamo la sezione che segue:

```
# AuthClass: the authorization class; currently only "Anonymous", "User",
# "System" (valid user belonging to group SystemGroup), and "Group"
# (valid user belonging to the specified group) are supported.
#
# AuthGroupName: the group name for "Group" authorization.
#
# Order: the order of Allow/Deny processing.
#
# Allow: allows access from the specified hostname, domain, IP address,
# network, or interface.
#
```

```
# Deny: denies access from the specified hostname, domain, IP address,
# network, or interface.
#
# Both "Allow" and "Deny" accept the following notations for addresses:
#
#   All
#   None
#   *.domain.com
#   .domain.com
#   host.domain.com
#   nnn.*
#   nnn.nnn.*
#   nnn.nnn.nnn.*
#   nnn.nnn.nnn.nnn
#   nnn.nnn.nnn.nnn/mm
#   nnn.nnn.nnn.nnn/mmm.mmm.mmm.mmm
#   @LOCAL
#   @IF(name)
#
# The host and domain address require that you enable hostname lookups
# with "HostNameLookups On" above.
#
# The @LOCAL address allows or denies from all non point-to-point
# interfaces. For example, if you have a LAN and a dial-up link,
# @LOCAL could allow connections from the LAN but not from the dial-up
# link. Similarly, the @IF(name) address allows or denies from the
# named network interface, e.g. @IF(eth0) under Linux. Interfaces are
# refreshed automatically (no more than once every 60 seconds), so
# they can be used on dynamically-configured interfaces, e.g. PPP,
# 802.11, etc.
#
# Encryption: whether or not to use encryption; this depends on having
# the OpenSSL library linked into the CUPS library and scheduler.
#
# Possible values:
#
#   Always      - Always use encryption (SSL)
#   Never       - Never use encryption
#   Required    - Use TLS encryption upgrade
#   IfRequested - Use encryption if the server requests it
#
# The default value is "IfRequested".
#
<Location />
Order Deny,Allow
Deny From All
Allow From 127.0.0.1
</Location>
```

Alla voce `<Location />` aggiungiamo una riga contenente l'indirizzo IP del client (ad esempio 192.168.0.3) per autorizzarlo, di modo che risulti:

```
<Location />
Order Deny,Allow
Deny From All
Allow From 127.0.0.1
Allow From 192.168.0.3
</Location>
```

Salviamo e riavviamo da console il servizio CUPS col comando `/etc/rc.d/rc.cups restart`. Se il riavvio ha funzionato, leggeremo il messaggio «scheduler restarted».

Nel client, lanciamo CUPS con la solita procedura e avviamo la configurazione di una nuova stampante, inserendo la descrizione «remote» anziché «locale». Come tipo di connessione scegliamo IPP (Internet Printing Protocol) e in DEVICE URL inseriamo la riga:

```
ipp://192.168.0.2:631/printers/hp_deskjet_6122
```

Scelto anche il driver opportuno e terminata la configurazione, nella schermata finale dovremmo vedere la stampante già connessa dall'indirizzo `http://192.168.0.2:631/printers/hp_deskjet_6122`. A questo punto è sufficiente stampare la pagina di prova per verificare che tutto funzioni correttamente.

9.1.5 Condividere una stampante tra GNU/Linux e Windows

La condivisione di una stampante fra due computer di cui l'uno monta un sistema operativo GNU/Linux e l'altro Windows è possibile anche senza fare ricorso a Samba. Unica condizione è che la macchina Windows supporti il protocollo IPP: Windows XP è già approntato per questo servizio, mentre le versioni precedenti richiedono l'installazione di un apposito driver aggiuntivo.

Primo caso: stampante installata su macchina GNU/Linux. Presupponiamo che la stampante vi sia già stata configurata come indicato sopra per una condivisione fra macchine GNU/Linux.

In Windows, andiamo nel menu **Stampanti e fax** e lanciamo il wizard **Aggiungi stampante**. Nella finestra **Stampante locale o di rete** scegliamo la voce Stampante di rete o stampante collegata a un altro computer. Nella successiva finestra, **Specificare una stampante**, scegliamo la voce Connetti a una stampante in Internet o della rete domestica o aziendale; inseriamo quindi l'URL della stampante indicando un indirizzo di questo tipo:

```
http://nome_server:631/printers/nome_stampante
```

dove `nome_server` è il nome della macchina GNU/Linux in cui è installata fisicamente la stampante e `nome_stampante` è il nome attribuito alla stampante in CUPS, ad esempio:

```
http://darkstar:631/printers/hp6122
```

Si aprirà quindi una nuova finestra che consente di aggiungere i driver. Una volta selezionati i driver corretti, comparirà un'ultima finestra, **Stampante predefinita**, che consente di impostare o meno la stampante appena aggiunta come predefinita. Sarà quindi possibile stampare una pagina di prova per verificare la felice riuscita dell'operazione.

Secondo caso: stampante installata su macchina Windows. È sufficiente installare la stampante anche sulla macchina GNU/Linux da CUPS, scegliendo alla voce Device il protocollo IPP e inserendo quindi l'indirizzo del server di stampa Windows, secondo la sintassi sopra ricordata.

9.2 Installare uno scanner USB

È possibile verificare se lo scanner è supportato da GNU/Linux (non tutti lo sono, soprattutto vecchi modelli) all'indirizzo: <http://www.sane-project.org>.

9.2.1 Kernel 2.4

Per installare uno scanner USB su Slackware è necessario innanzitutto che lo scanner venga riconosciuto dal sistema. Esso deve essere rilevato già all'avvio del programma, se è stato abilitato HOTPLUG. Possiamo accertarcene col comando: `sane-find-scanner`, che ci restituirà un output del tipo:

```
found USB scanner ... now attached to scanner0
```

oppure

```
found SCSI scanner ...
```

a seconda dello scanner installato. Si può poi andare nella directory `/etc/sane.d`, individuare il file *marca_dello_scanner.conf* che corrisponde alla marca del nostro scanner (ad esempio `epson.conf` oppure `hp.conf`), e lasciare decommentata solo la riga relativa alla porta del dispositivo già suggeritaci dal comando `sane-find-scanner`; nel caso di uno scanner USB, sarà `/dev/usb/scanner0` oppure `/dev/usbscanner0`. Infine dovremo controllare che sia decommentata la riga relativa alla marca dello scanner in `/etc/sane.d/dll.conf`. A questo punto, possiamo verificare se lo scanner è funzionante col comando: `scanimage -L`. Se l'output è qualcosa di simile a:

```
device `epson:libusb:004:002' is a Epson Perfection1200 flatbed scanner
```

potremo usare, ma solo da root, tutti i programmi di scansionamento.

Per far sì che anche gli utenti possano usare lo scanner, una prima strategia, pragmaticamente efficace, consiste nel modificare i permessi di `/dev/usb/scanner0` oppure di `/dev/usbscanner0` col comando `chmod 777`. In questo modo tuttavia (oltre al fatto che la modifica apportata non sarebbe stabile se si adotta udev) si riceverà all'avvio un (peraltro innocuo) messaggio di errore:

```
chown: 'root:scanner': invalid group
```

È perciò soluzione migliore gestire lo scanner inserendo gli utenti autorizzati nel gruppo `scanner`, che da Slackware 10.2 esiste già (nelle versioni precedenti era invece necessario crearlo a mano col comando: `groupadd scanner`) mediante il comando: `usermod -G scanner nome_utente` ed eventualmente attribuendo il dispositivo di scansionamento al gruppo `scanner` col comando: `chgrp scanner /dev/usb/scanner0`.

9.2.2 Kernel 2.6

Se è stato compilato il kernel 2.6, ci si trova davanti a una diversa gestione delle porte USB, che non è più governato attraverso i dispositivi di `/dev`, bensì da **libusb**. In questo caso è facile verificare che in `/etc/sane.d/epson.conf` sono commentate le righe:

```
usb /dev/usb/scanner0
```

e


```
usb /dev/usbscanner0
```

La voce

```
usb
```

è invece già presente e attiva, e non serve aggiungere nulla. Il file `/etc/hotplug/usb.usermap` contiene già tutti i dati rilevanti (VENDOR e PRODUCT) di un gran numero di scanner supportati dal progetto Sane. Queste righe abilitano il riconoscimento via libusb dello scanner passandogli VENDOR e PRODUCT. Solo se lo scanner risultasse sconosciuto, ne dovremmo aggiungere la localizzazione su libusb suggeritaci prima da `sane-find-scanner` con la sintassi `usb <product ID> <device ID>`, ad esempio:

```
usb libusb:004:002
```

A questo punto possiamo usare tutte le applicazioni dello scanner, ma solo da root. Come renderne possibile l'utilizzo anche da utente? Per evitare ogni volta la scomoda procedura di conferire i permessi da root e quindi rientrare come utente, si può automatizzare il tutto (anche se in realtà Slackware è già predisposta a ciò e richiede solo un intervento minimo). Come premessa è necessario aver abilitato hotplug. Slackware contiene in `/etc/hotplug/usb` lo script `libusbscanner`, il quale di default conferisce a tutti gli utenti del gruppo «scanner» i permessi di utilizzo (chmodando il dispositivo al valore 660 e attribuendone la proprietà a `root:scanner`); basta pertanto aggiungere al gruppo «scanner» i nomi degli utenti autorizzati. Nello stesso script, le ultime righe conferiscono a tutti gli utenti, indipendentemente dal gruppo di appartenenza, la possibilità di usare lo scanner (chmodando il dispositivo al valore 666). Tali righe, di default, sono commentate. Basterà decommentarle, anche se allargare indiscriminatamente permessi potrebbe introdurre delle falle nella sicurezza del sistema.

9.2.3 Condividere uno scanner

È possibile condividere uno scanner fra diversi computer, in modo che tutti i client possano gestire i programmi di scansione anche se esiste un solo scanner, connesso al computer server. Se ne occupa il demone saned. Per usare questa funzione, va innanzitutto verificato che in `/etc/sane.d/dll.conf` sia decommentata la riga:

```
net
```

(che comunque in Slackware lo è già).

Per configurare il servizio sul lato server, innanzitutto aggiungiamo in `/etc/services` la riga seguente:

```
sane 6566/tcp #Sane network scanner daemon
```

Quindi, visto che Slackware è basato su `inetd` e non su `xinetd`, aggiungiamo in `/etc/inetd.conf` la seguente riga:

```
sane stream tcp nowait root /usr/sbin/tcpd /usr/sbin/saned
```

La prima stringa indica il nome del servizio (`sane`), la seconda il tipo di socket (`stream`), la terza il protocollo (`tcp`), la quarta indica di attivare il servizio immediatamente alla richiesta del client, la quinta il proprietario del processo (`root`), la sesta e la settima il path dei processi.

A questo punto, dopo aver riavviato `inetd` (col comando: `/etc/rc.d/rc.inetd restart`), avremo il servizio `saned` in ascolto sulla porta 6566, cosa che si può verificare con `telnet` dando il comando: `telnet localhost 6566`. Se l'output è:

```
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
```

significa che la porta 6566 su cui è attivato saned è effettivamente in ascolto.

Infine aggiungiamo in `/etc/sane.d/saned.conf` l'indirizzo dei client autorizzati a fruire del servizio (cioè l'indirizzo IP o anche il nome, se è attivato il servizio di risoluzione dei nomi):

```
192.168.0.3
```

Nei client è sufficiente decommentare la voce `localhost` e aggiungere in `/etc/sane.d/net.conf` una riga che definisce il server (anche in questo caso l'IP può essere sostituito col nome del server) con la sintassi `net:IP`, ossia:

```
net:192.168.0.2
```

Dal client potremo verificare se il dispositivo di scansione è stato correttamente individuato col comando `scanimage -L`, che fornirà un output del tipo:

```
device `net:192.168.0.2:epson:libusb:004:002' is a Epson Perfection1200 flatbed scanner
```

Se insorgono degli errori, o la connessione client/server non ha luogo, ci si può rendere conto di quello che non funziona col comando: `SANE_DEBUG_NET=128 scanimage -L`.

Saned è in grado anche di consentire l'accesso allo scanner solo agli utenti di un determinato gruppo, o solo in base a una password, ma qui prescindiamo da queste ulteriori configurazioni. Infine, può essere usato anche per una condivisione dello scanner fra computer GNU/Linux e computer Windows. Esistono in questo caso delle apposite utilità: un programma open source è reperibile all'indirizzo: <http://sanetwain.ozuzo.net>.

9.3 Installare una webcam

9.3.1 La webcam

Il funzionamento della webcam USB implica che nel sistema siano presenti i driver corretti (e che sia attivato hotplug). Il kernel di GNU/Linux supporta svariati modelli di webcam; in caso contrario bisognerà cercare in rete se ne esistano. Siti di riferimento (in cui trovare anche altri link) sono:

<http://www.linux-usb.org>

<http://alpha.dyndns.org/ov511>

<http://www.smcc.demon.nl/webcam/release>

In ogni caso, è di prammatica il ricorso ai motori di ricerca.

Innanzitutto ci accerteremo che la webcam sia riconosciuta, inserendola e controllando l'esito col comando: `tail /var/log/messages`, che dovrebbe offrire, fra l'altro, un output del tipo:

```
Jul 11 17:23:48 darkstar kernel: ohci_hcd 0000:00:03.0: remote wakeup
Jul 11 17:23:48 darkstar kernel: usb 2-1: new full speed USB device using address 3
Jul 11 17:23:48 darkstar kernel: drivers/usb/media/ov511.c: USB 0V511+ video device
found
Jul 11 17:23:48 darkstar usb.agent[4235]: ... no modules for USB product 5a9/a511/100
```

```
Jul 11 17:23:48 darkstar kernel: drivers/usb/media/ov511.c: model: Creative Labs WebCam 3
Jul 11 17:23:49 darkstar kernel: drivers/usb/media/ov511.c: Sensor is an OV7620
Jul 11 17:23:50 darkstar kernel: drivers/usb/media/ov511.c: Device at usb-0000:00:03.0-1 registered to minor 0
Jul 11 17:23:50 darkstar udev[4273]: configured rule in '/etc/udev/rules.d/udev.rules' at line 103 applied, added symlink '%k'
Jul 11 17:23:50 darkstar udev[4273]: configured rule in '/etc/udev/rules.d/udev.rules' at line 103 applied, 'video0' becomes 'v4l/video%n'
Jul 11 17:23:50 darkstar udev[4273]: creating device node '/dev/v4l/video0'
```

Il comando `lsusb` dovrebbe invece dare un output del tipo:

```
Bus 004 Device 002: ID 04b8:0104 Seiko Epson Corp. Perfection 1200
Bus 004 Device 001: ID 0000:0000
Bus 003 Device 002: ID 03f0:3404 Hewlett-Packard
Bus 003 Device 001: ID 0000:0000
Bus 002 Device 003: ID 05a9:a511 OmniVision Technologies, Inc. OV511+ WebCam
Bus 002 Device 001: ID 0000:0000
Bus 001 Device 001: ID 0000:0000
```

dove quello che ci interessa non è né lo scanner (Epson), né la stampante (Hewlett-Packard) ma la terza voce. In alcuni casi la webcam è riconosciuta come una generica periferica di acquisizione ottica, insomma più o meno come uno scanner, cosicché la troveremo anche dando il comando: `sane-find-scanner`, il quale ci restituirà fra l'altro un output del tipo:

```
found USB scanner (vendor=0x04b8 [EPSON], product=0x0104 [Perfection1200]) at libusb:004:002
found USB scanner (vendor=0x05a9, product=0xa511) at libusb:002:003.
```

9.3.2 Gnomemeeting

L'installazione di Gnomemeeting è molto lineare, anche se richiede alcune librerie di Gnome non presenti in Slackware per poter funzionare. Se sono stati installati i driver corretti, il programma funzionerà da subito, una volta effettuata una semplice configurazione guidata.

Capita talora che Gnomemeeting accusi una cattiva configurazione di Gconf, non riuscendo a trovare la chiave di registro `gconf_test_age` e interrompa così il processo iniziale di settaggio. Questo problema è dovuto esclusivamente alla cattiva compilazione del pacchetto. Per rimediare, è comunque sufficiente lanciare da console il comando: `gnomemeeting-config-tool --install-schemas` e quindi confermare l'installazione. Compiuta anche questa operazione, lanciando Gnomemeeting si avvierà finalmente il Configuration Druid per settare la webcam, ormai pronta per l'uso.

La più recente versione di Gnomemeeting prende di nome di Ekiga; si tratta di un software fortemente integrato in Gnome, per cui, a meno di non voler installare una nutrita serie di dipendenze, dovrebbe essere fatto girare nel suo ambiente nativo.

9.4 Installare una macchina fotografica digitale

Ormai le macchine fotografiche digitali stanno scardinando il primato della pellicola. A fronte del fatto che non sempre la qualità delle immagini è buona come quella offerta dalle macchine tradizio-

nali (ma vi sono modelli sempre più perfezionati e ad altissima definizione), esse offrono la possibilità di manipolare in maniera veloce e precisa le fotografie. La gestione di questo tipo di dispositivi sotto GNU/Linux non presenta dei grossi problemi, anche se in Slackware può richiedere una certa iniziativa da parte dell'utente.

Molte macchine fotografiche digitali supportano sia lo standard USB Mass Storage, per cui sono viste come semplici dischi rimovibili USB, sia lo standard PTP (Picture Transfer Protocol), che consente una gestione avanzata del dispositivo. Basta modificarne il setup per usare il primo o il secondo protocollo.

9.4.1 Le macchine fotografiche digitali come dischi USB

Nel caso si ricorra all'USB Mass Storage, la fotocamera connessa viene riconosciuta come un disco rimovibile USB (`/dev/sdn`), cioè né più né meno che come un pendrive. Varranno quindi le procedure descritte a suo luogo; si può aggiungere una riga a `/etc/fstab`, oppure ricorrere al servizio di automount. Per visualizzare e copiare o cancellare le foto si monta questo nuovo disco e ci si comporta col suo contenuto come con dei dati di un normale disco. Il ricorso a una regola di udev del tipo:

```
BUS="scsi",SYSFS(MANUFACTURER)="marca",SYSFS(product)="modello",NAME="usb_camera%n"
```

potrebbe facilitarne la gestione.

Una interessante caratteristica delle macchine fotografiche digitali consiste nell'utilizzo dello standard Exif, che consente di memorizzare all'interno del file di immagine anche dati relativi allo scatto (l'apertura del diaframma, la lunghezza focale, l'eventuale uso del flash, ecc.). Questi dati possono essere visti non solo, com'è ovvio, dal menù della macchina fotografica, ma anche dal computer con i programmi che supportano tale standard. Citiamo qui:

Gqview: si tratta di un visualizzatore veloce e stabile, che possiede alcune interessanti caratteristiche come l'anteprima delle immagini, lo zoom e la possibilità di ricorrere a editor esterni. Non è incluso tra i pacchetti ufficiali di Slackware e si può scaricare all'indirizzo: <http://gqview.sourceforge.net>.

Gexif: un frontend grafico per gqview. Anch'esso non è incluso tra i pacchetti ufficiali di Slackware e si può scaricare all'indirizzo: http://sourceforge.net/project/showfiles.php?group_id=12272. Esso consente non solo la visualizzazione, ma anche la modifica dei dati.

gThumb: è un visualizzatore avanzato per Gnome, ricchissimo di funzioni come la ricerca, la riorganizzazione e l'aggiornamento delle immagini, la loro manipolazione in diversi formati grafici.

GIMP è in grado di conservare tali valori se si salva in .jpg, ma non di visualizzarli e, conseguentemente, nemmeno di modificarli.

Konqueror visualizza il contenuto della macchina fotografica scrivendo semplicemente nella barra degli indirizzi `camera:/. Richiede però che sia abilitata Kamera (vedi sotto).`

9.4.2 Il protocollo PTP

Il protocollo PTP è un standard dovuto alla International Imaging Industry Association atto a trasferire immagini da una fotocamera a un computer senza la necessità di ricorrere a driver specifici. Per la gestione, è però necessario avere installate nel sistema le `libgphoto2`, che consentono di interfacciare oltre cinquecento macchine fotografiche digitali, ma che non sono presenti fra i pacchetti ufficiali di Slackware. Le `libgphoto2` fanno sì che `hotplug` attribuirà la gestione dei dispositivi fotografici a `root:root`; è perciò conveniente redigere uno script `doinst.sh` che crei un gruppo `camera`, per consentirne l'uso agli utenti che vi fanno parte. Non si dovrà fare altro che inserirne il nome di tali utenti del file `/etc/group`, come si è fatto per lo scanner.

Dopo queste operazioni fondamentali, è vantaggioso installare ulteriori pacchetti, per poter fruire di tutti i vantaggi delle `libgphoto2`, nessuno dei quali rientra in quelli ufficiali di Slackware. Menzioniamo qui:

gphoto2 = una serie di applicazioni per la gestione delle fotocamere.

kamera = l'io_slave di Konqueror. In realtà, Kamera è un'utilità che fa parte di Kdegraphics. Poiché abilitare questa funzione richiede che siano già installate le `libgphoto2`, essa non è presente nella versione ufficiale di KDE offerta da Slackware. Bisognerà pertanto ricompilare Kdegraphics inserendo nel configure anche la voce: `--with-kamera`. È sufficiente modificare in tal senso lo SlackBuild presente in `/source/kde/kdegraphics`.

digikam = interfaccia a `gphoto2` per KDE. Esso ha la semplice funzione di visualizzatore, ma può essere integrato con un pacchetto di plugin che ne amplia in misura accentuata le funzionalità.

digikamimage-plugins = serie di plugin per Digikam che consentono, fra l'altro, di rinominare gruppi di immagini, di cambiarne il formato, di trasformarle in Video CD, di archivarle su CD, di creare una galleria di immagini per il web, di correggere gamma, luminosità e contrasto, di preparare le immagini per la stampa, di inviarle via e-mail e altro ancora.

Lanciato Digikam, dal menu **Fotocamera** si seleziona **Aggiungi fotocamera**, e si può cliccare sul pulsante **Autorilevamento**. Se la macchina fotografica è tra quelle direttamente supportate, verrà identificata e ne verranno indicate le caratteristiche (marca e modello), altrimenti sarà aggiunta come un generico dispositivo PTP. Dopo di che si potrà scaricare il contenuto della macchina su disco fisso.

Per gli utenti che hanno installato Gnome, esiste un'altra applicazione di base per la gestione delle fotocamere digitali, e cioè **gtkam**, un frontend per le `libgphoto2`.

Capitolo 10

Avviare il sistema e creare un dual boot

10.1 Il dual boot

Molti utenti di GNU/Linux, per la fase di migrazione, per scelta o per necessità, dovranno mantenere sul loro disco fisso un altro sistema operativo, che può essere Windows o un'altra distribuzione di GNU/Linux. Per accedere a entrambi è necessario riconfigurare il bootloader di Slackware, e cioè Lilo. Va precisato che ogni sistema operativo ha comunque bisogno di un bootloader per essere caricato; anche Windows ha infatti il suo bootloader, che fa riferimento al file `C:\boot.ini`. Di conseguenza, anche se Slackware fosse l'unico del nostro disco fisso, continuerebbe a essere necessaria la presenza di Lilo (o di Grub).

10.2 Lilo

10.2.1 Il bootloader Lilo

Storicamente, Lilo (Linux LOader) è il primo bootloader di GNU/Linux. Superati i limiti tecnici delle prime versioni (soprattutto il fatto che esso non leggeva oltre i primi 1024 cilindri del disco fisso) e la differenza fra il modo in cui talvolta il BIOS e il kernel leggono la geometria del disco fisso, esso è attualmente un bootloader universale, in grado di caricare un vastissimo numero di sistemi operativi diversi. Le versioni più recenti sono in grado di identificare automaticamente non solo le partizioni DOS, ma anche quelle NTFS. Inoltre Lilo gestisce una serie di parametri opzionali che gli vengono passati in fase di boot. Con opportune patch al kernel, può essere abbellito con accattivanti bootsplash.

10.2.2 Modalità di installazione di Lilo

Slackware ci consente di configurare facilmente Lilo mediante il menu pseudo-grafico **liloconfig** (che si lancia da shell col comando: `liloconfig`) Questa utilità presenta due modalità: **simple**, che pone alcune semplici domande, ed **expert**, che richiede un po' di lavoro in più.

Modalità simple

È la modalità più comoda per gli utenti inesperti e presenta un'ottima capacità di riconoscimento dei dispositivi. Questa modalità però riconosce in genere Windows e, per GNU/Linux, la sola partizione in cui è presente Slackware. Se abbiamo sul nostro computer altre distribuzioni GNU/Linux, queste dovranno essere aggiunte successivamente a mano. Tale modalità chiede di impostare i seguenti campi:

frame buffer = attivazione del frame buffer per la console. La modalità standard (vga normale) fa vedere all'avvio dei caratteri piuttosto grandi, poco definiti (e senza logo); usando il framebuffer appropriato alla nostra scheda video invece i caratteri sono più piccoli e appare anche l'immagine di un pinguino sullo schermo (sebbene lo scorrimento del testo sia in questo caso leggermente più lento).

parametri opzionali (e precisamente la voce `OPTIONAL LILO append="<kernel parameters>"` LINE) = aggiunta di parametri facoltativi alla riga **append**, come l'abilitazione dell'emulazione SCSI, oppure del tipo `ramdisk, read-only, read-write, root e vga`.

destinazione = scelta della sede dove installare Lilo.

Modalità expert

La modalità expert consente una configurazione più fine, cosicché non dovrebbe esserci bisogno di applicare successivamente ritocchi al file `/etc/lilo.conf`. Chiede di impostare i seguenti campi:

begin = propone di aggiungere degli header, e cioè parametri opzionali per il kernel.

frame buffer = attivazione del frame buffer per la console.

target = la sede dove installare Lilo.

timeout = il tempo che deve passare perché si carichi il sistema di default.

add a Linux partition = consente di aggiungere le partizioni GNU/Linux già riconosciute da Lilo. Si seleziona la partizione e le si attribuisce un nome a piacere. Non serve, com'è ovvio, aggiungere a Lilo le partizioni non bootabili (come una partizione destinata a fare da backup di dati e priva di sistema operativo).

add a Windows partition = consente di aggiungere le partizioni Windows già riconosciute da Lilo. Si seleziona la partizione e le si attribuisce un nome a piacere.

Tutte queste configurazioni vengono riportate in `/etc/lilo.conf`, un file di puro testo che può essere modificato con un editor qualsiasi.

10.2.3 La sintassi di Lilo

Il file di configurazione di Lilo è `/etc/lilo.conf`. Diamo conto delle sue principali voci e opzioni.

In # Start LILO global section troviamo le voci seguenti:

`boot = /dev/hda`, che ci dice da dove viene effettuato il boot del sistema e dove installeremo Lilo (si tratta del disco fisso, non di una partizione).

`default = linux`, che determina il sistema operativo di default.

`message = /boot/boot_message.txt`, riferimento al file dov'è situato il messaggio nel prompt di apertura, che possiamo modificare a nostro piacimento con un editor di testo.

`prompt = forza`, senza la pressione di alcun tasto, l'apertura di un prompt che permette di scegliere il sistema operativo da caricare e di inserire parametri opzionali.

`timeout = 2000`, che determina quanti secondi devono passare perché scompaia il prompt e si avvii il sistema di default (il tempo è indicato in decimi di secondo). Il default in Slackware è 2000 decimi di secondo, e cioè 20 secondi. Attenzione a non impostare questa voce a 0, altrimenti non ci sarà più la possibilità di scegliere i sistemi di avvio alternativi, a meno che non si intenda disabilitare in qualche modo il dual boot a caricare immediatamente il kernel. In ogni caso, è possibile far comparire la finestra del prompt premendo il tasto **Tab** al momento del boot.

Altre utili opzioni aggiuntive sono:

`linear` = suggerisce a Lilo di non considerare la geometria del disco fisso, o meglio di generare indirizzi di settore lineari anziché del tipo settore/testina/cilindro. È utile quando il BIOS e il kernel vedono una diversa geometria del disco stesso. In questo caso Lilo non conta il numero dei cilindri.

`compact` = abilita la compattazione della mappa (il file di mappa è `/boot/map`). Ciò unirà le richieste di lettura di settori consecutivi, velocizzando l'avvio del sistema, soprattutto da floppy.

lba32 = genera degli indirizzi a blocchi logici da 32 bit invece che considerare gli indirizzi in base a cilindro/testina/settore. Ciò consente di avviare un sistema dotato di un disco fisso con più di 1024 cilindri. Dalla versione 22, questa è l'impostazione di default (che perciò non serve precisare) e rimedia a un noto limite delle versioni precedenti.

map = serve a rimappare i dispositivi se, ad esempio nel caso che si aggiunga un disco fisso al sistema, Windows non si colloca più nella prima partizione del primo disco.

È di conseguenza possibile far partire comunque Windows, in certo modo «ingannandolo» mediante Lilo. La sezione pertinente apparirà come segue:

```
other = /dev/hd $xn$ 
map-drive = 0x80
to = 0x81
map-drive = 0x81
to = 0x80
label="Windows"
table = /dev/hda
```

dove, al solito, x è la lettera che indica il disco fisso e n la partizione dove risiede Windows.

Volendo, è anche possibile aggiungere la riga `password=password`, di modo che l'accesso sia limitato.

In # Linux bootable partition config begins troviamo:

`image = /boot/vmlinuz`, che indica il kernel di GNU/Linux che dev'essere caricato all'avvio e che si trova in /boot.

`root=hd xn` = indica la partizione del disco fisso dove risiede la directory principale di Slackware (in questo caso, «root» non indica perciò l'utente root) e dove andranno a installarsi i file di configurazione di Lilo.

`label =` indica semplicemente il nome del programma che incontreremo nel messaggio di boot.

`read-only =` fa in modo che Lilo istruisca il kernel perché carichi il file system di root in sola lettura. Esso verrà rimontato in lettura e scrittura più tardi, durante il processo di boot.

In # DOS bootable partition config begins troviamo:

`other = /dev/hda1`, che indica la partizione in cui è situato il sistema operativo alternativo

`label = windows`, che indica ancora una volta il nome a piacere che intendiamo attribuire alla partizione (e che ritroveremo nel messaggio iniziale).

`table = /dev/hda` = identifica la collocazione della descriptor table, il dispositivo da dove viene effettuato il boot del sistema.

Scheda n. 20

Un esempio di /etc/lilo.conf

Ecco un caso di /etc/lilo.conf relativo allo stesso sistema di cui, alla Scheda n. 14, abbiamo riportato sopra il file /etc/fstab:

```
# LIL0 configuration file
# generated by 'liloconfig'
#
# Start LIL0 global section
boot = /dev/hda
```

```
message = /boot/boot_message.txt
prompt
timeout = 50
default = Slackware
# Override dangerous defaults that rewrite the partition table:
change-rules
  reset
# VESA framebuffer console @ 1024x768x256
vga = 773
# Normal VGA console
# vga = normal
# VESA framebuffer console @ 1024x768x64k
# vga=791
# VESA framebuffer console @ 1024x768x32k
# vga=790
# VESA framebuffer console @ 1024x768x256
# vga=773
# VESA framebuffer console @ 800x600x64k
# vga=788
# VESA framebuffer console @ 800x600x32k
# vga=787
# VESA framebuffer console @ 800x600x256
# vga=771
# VESA framebuffer console @ 640x480x64k
# vga=785
# VESA framebuffer console @ 640x480x32k
# vga=784
# VESA framebuffer console @ 640x480x256
# vga=769
# End LILO global section
# Windows bootable partition config begins
other = /dev/hda1
  label = Windows
  table = /dev/hda
# Windows bootable partition config ends
# Linux bootable partition config begins
image = /boot/vmlinuz
  root = /dev/hda2
  label = Slackware
  read-only
image = /mnt/slack/boot/vmlinuz
  root = /dev/hdb3
  label = Slack
  read-only
image = /mnt/debian/boot/vmlinuz
  initrd = /mnt/debian/boot/initrd.img-2.4.27-1-386
  root = /dev/hdb4
  label = Debian
  read-only
```

```
# Linux bootable partition config ends
```

10.2.4 La password di Lilo

Per proteggere l'accesso al sistema, è possibile aggiungere alle password di login dei vari utenti una password di Lilo. La password di Lilo è di due tipi: può essere generale, ossia richiesta a ogni accesso (password), oppure entrare in funzione solo se vengono specificati a Lilo parametri speciali sulla riga di comando (restricted). Questo secondo scenario è importante perché qualcuno potrebbe bloccare l'avvio del kernel per passargli alcuni parametri aggiuntivi e cercare di ottenere la shell di root. Nel primo caso (avvio riservato) si edita il file `/etc/lilo.conf` aggiungendo la riga:

```
password = password
```

In tal modo solo l'amministratore può avviare il sistema. Nel secondo caso (avvio libero, ma protetto), chiunque può avviare il computer, ma solo l'amministratore è in grado di passare parametri aggiuntivi al kernel. A questo fine si edita il file `/etc/lilo.conf` aggiungendo, oltre alla riga `password`, la riga:

```
restricted
```

In tal modo il sistema, senza la password, si avvierà solo in modalità multiutente (runlevel 4). Per chi desidera una maggiore sicurezza, c'è anche un sistema più complicato. Si tratta di installare Lilo sia nell'MBR del disco fisso, sia nel settore di avvio della partizione di root, in modo che il primo richiami il secondo. La configurazione di questo primo Lilo dovrà essere come segue:

```
# /etc/lilo.hda
boot = /dev/hda
message = /boot/message.txt
prompt
password = password1
other = /dev/hd $\alpha$ n
label = Linux
```

Se la password fornita al primo Lilo è corretta, si avvia un sistema operativo generico (definito infatti «other», mentre il kernel viene in effetti avviato solo nella seconda fase di avvio) presente nella partizione di root. Questo Lilo avvierà automaticamente Slackware senza parametri, i quali, per essere aggiunti, necessiteranno della seconda password (ovviamente diversa dalla prima). Al boot, un messaggio avvertirà di contattare l'amministratore per eventuali problemi. Questo meccanismo serve quando vi siano più utenti, non in un computer domestico in cui facilmente root coincide con l'utente principale. La configurazione del secondo Lilo (dove si potranno aggiungere i parametri aggiuntivi di append) dovrà essere come segue:

```
boot = /dev/hda1
vga = normal
root = /dev/hda1
password = password2
restricted
read-only # Non-UMSDOS filesystems should be mounted read-only for checking
```

Se si opta per una simile configurazione, si dovrà installare prima il Lilo della partizione di root e solo in seguito l'altro, giacché Lilo, se non trova un sistema operativo nelle partizioni indicata da «other», rifiuta di installarsi. Perciò i comandi saranno:

```
lilo -C /etc/hda1
```

```
lilo -C /etc/hda
```

o comunque si chiamino le partizioni. Se si imposta la password di Lilo, poiché essa viene scritta in chiaro (e non in maniera criptata) nel file `/etc/lilo.conf`, quest'ultimo dovrà avere i permessi 600 (leggibile e scrivibile solo da root), o altri utenti saranno in grado di leggere le password. Infine, si può rendere imm modificabile il file di configurazione di Lilo, mediante il comando `chattr`, il che previene anche modificazioni accidentali. Solo root potrà rimuovere questo attributo. Il comando è: `chattr +i /etc/lilo.conf`. L'opzione `+i` aggiunge l'immodificabilità, mentre per rimuoverla si passerà l'opzione `-i`.

10.2.5 Aggiungere Windows

Per poter avviare anche Windows dovremo aggiungere una sezione del tipo:

```
# DOS bootable partition config begins
other = /dev/hda1
label = Windows
table = /dev/hda
# DOS bootable partition config ends.
```

Le versioni più recenti di Lilo riconoscono non solo le partizioni DOS (FAT e FAT32), ma anche NTFS, per cui in genere esse vengono aggiunte automaticamente, rendendo superflua la configurazione manuale.

10.2.6 Riconfigurare Lilo

Dopo aver modificato `/etc/lilo.conf`, lanceremo il comando `lilo` per far sì che la nuova configurazione venga scritta nell'MBR. In realtà, per maggior cautela, possiamo lanciare dapprima `lilo -t` (`t` = trial), che ci mostra che cosa accadrebbe lanciando Lilo, e, se non riceviamo messaggi di errore e constatiamo che i sistemi operativi che vogliamo sono stati individuati correttamente, `lilo -v` (`v` = verbose), che installa Lilo mostrandoci il progresso della scrittura. Non resta che riavviare il sistema.

10.2.7 Il multiboot

È naturalmente possibile installare più sistemi operativi, e non solo due. La procedura per il multiboot è comunque la stessa del dual boot. Ipotizziamo di avere un'altra partizione con una seconda distribuzione di GNU/Linux, diciamo Debian, installata sulla quarta partizione del primo disco fisso (e dunque in `/dev/hda4`). Nella sezione **# Linux bootable partition config begins** prima della riga **# Linux bootable partition config ends** dovremo aggiungere le righe: a) `image`, b) `root`, c) `label` e d) `read-only` dell'ulteriore sistema operativo.

In a) collochiamo il riferimento al kernel di Debian da caricare. Posto che da Slackware lo vediamo in `/mnt/debian`, la riga risulterà:

```
/mnt/debian/boot/vmlinuz
```

In b) indichiamo la partizione della directory root di Debian, nel nostro esempio `/dev/hda4`. In c) indichiamo il nome che ci apparirà nel messaggio di avvio, che potrebbe essere per l'appunto Debian.

Poiché, inoltre, alcune distribuzioni vogliono alcune indicazioni supplementari come `initrd` (per caricare i moduli opportuni in fase di avvio del sistema), esse andranno aggiunte nella medesima sezione. Questa apparirà dunque come segue:

```
image = /mnt/debian/boot/vmlinuz-2.4.27-1-386
root = /dev/hda4
label = Debian
initrd = /mnt/debian/boot/initrd.img-2.4.27-1-386
read-only
```

10.2.8 Reinstallazione di Lilo

Nel caso, che talvolta si verifica, che durante l'installazione il programma non riesca a installare Lilo, o che per altri motivi lo abbiamo disinstallato o sovrascritto (ad esempio, installando Windows *dopo* Slackware), è necessario avviare Slackware da floppy oppure da CD. Nel caso si usi il floppy di avvio, esso sa in genere quale partizione caricare come root, per cui è sufficiente loggarsi come root e quindi lanciare in sequenza, come detto sopra, `lilo -t` (se si vuole effettuare un controllo) e `lilo -v`. Se invece si usa il CD1 di Slackware, sono da effettuare i seguenti passaggi:

- a) si fa il boot da CD, come per una normale installazione. Se il disco fisso non è IDE ma SATA, si passa al boot l'opzione `sata .i`; per le unità IDE non è necessario passare l'opzione `bare .i`, che viene scelta di default. Per casi diversi, si sceglie il kernel opportuno.
- b) si sceglie eventualmente la tastiera italiana
- c) si effettua il login come root
- d) si monta la partizione di root col comando: `mount -t file_system /dev/hd xn /mnt` (oppure, nel caso di dischi SATA, `/dev/sd xn` , dove `file_system` sarà EXT2, EXT3 o Reiserfs e `n` è il numero della partizione
- e) si dà il comando: `chroot /mnt`
- f) si lancia `lilo -t` (se si vuole effettuare un controllo) e quindi `lilo -v`.

Questa procedura va adottata anche nel caso si installi Windows dopo Slackware, perché il bootloader di Windows sovrascriverà quello di Slackware.

10.3 L'alternativa a Lilo: Grub

10.3.1 Il bootloader Grub

Negli `/extra` del CD 3, Slackware ci mette a disposizione un bootloader alternativo a Lilo, Grub (acronimo di Great Unified Boot loader), più recente e oggi adottato da numerose distribuzioni, perfettamente compatibile anche con Slackware. Grub è molto potente ed elastico, supporta un numero maggiore di opzioni rispetto a Lilo, fornisce per i parametri da passare al boot una comoda funzione di completamento automatico (col tabulatore, come si fa in console); inoltre, poiché punta direttamente ai file, a differenza di Lilo non va lanciato ogni volta che si modifica il kernel, il che rende maggiormente rapida la riconfigurazione del sistema. Un'analisi completa delle caratteristiche e funzionalità di Grub si può reperire al seguente indirizzo: <http://www.gnu.org/software/grub/manual/grub.pdf>.

10.3.2 Modalità di installazione

L'utilità `grubconfig`

Grub si installa semplicemente dal suo pacchetto, e in Slackware è dotato di una comoda utilità del tutto simile a `liloconfig`, che si chiama `grubconfig` e che si lancia da console col comando `grubconfig`. Il settaggio attraverso questa utilità è molto semplice. Essa prevede due modalità, una «simple» che fa quasi tutto il lavoro automaticamente, e una «expert» che richiede un po' di lavoro manuale in più. In ogni caso, in questa fase installeremo Grub in un floppy, per verificare che tutto funzioni. Possiamo sovrascrivere Lilo in un secondo momento, se decidessimo di sostituirlo col nuovo bootloader.

Modalità simple

È la modalità più comoda per gli utenti inesperti e presenta un'ottima capacità di riconoscimento dei dispositivi. L'unico difetto è che in questo modo Grub, rilevando tutte le partizioni, non distingue fra quelle avviabili e quelle che non lo sono e infatti tutte le aggiunge alla voce «Linux bootable partition». Sarà comunque sufficiente cancellarle manualmente da `/boot/grub/menu.lst`.

framebuffer: la modalità standard (vga normale) non attiva il framebuffer e perciò presenta all'avvio caratteri piuttosto grandi e fa andar perduta l'immagine del pinguino. In alternativa, selezionare una modalità maggiormente appropriata alla propria scheda.

partizione: la partizione dov'è installato Grub, che in linea di massima sarà quella di root di Slackware.

destinazione: dove installare Grub. Le alternative sono tre, come per Lilo: il superblocco della partizione di root, l'MBR del disco fisso, un floppy. Scegliamo questa opzione: il programma ci chiederà se vogliamo formattare il dischetto e, una volta pronto, scegliendo la voce «ready», passerà alla copiatura dei file. Se verificheremo il buon funzionamento del floppy, sempre dal menu di apertura potremo copiare in seguito Grub sull'MBR, installandolo stabilmente.

Modalità expert

Fornisce una configurazione più fine, anche se ulteriori modifiche possono sempre essere apportate manualmente, in un secondo tempo, al file `/boot/grub/menu.lst`.

header: richiede l'inserimento di eventuali header o parametri opzionali per il kernel, come un'immagine ramdisk iniziale (`initrd`).

framebuffer: come sopra.

destinazione: come sopra.

partizione: come sopra.

Aggiungere partizioni GNU/Linux: consente di aggiungere partizioni GNU/Linux specificandone il nome, la directory di boot e parametri opzionali aggiuntivi.

Aggiungere partizioni di altri tipo: consente di aggiungere partizioni non GNU/Linux (DOS, Windows, BSD, OS2 e HURD) specificandone il nome, la directory di boot e parametri opzionali aggiuntivi.

Infine installa Grub dove abbiamo scelto di collocarlo, come nella modalità simple.

10.3.3 La sintassi di Grub

Il file di configurazione di Grub è `/boot/grub/menu.lst`, un file di solo testo che può essere modificato a mano.

La sintassi di Grub è tuttavia leggermente diversa da quella di Lilo. Innanzitutto le partizioni sono indicate con sigle e numeri che partono da 0 e non da 1. Così, il disco fisso master risulta essere `hd0` e lo slave secondario `hd1` e le rispettive partizioni vengono indicate con dei numeri separati dal nome della partizione da una virgola senza spazi, anche in questo caso partendo da 0. Dunque, la prima partizione del disco fisso master sarà `hd0,0` e la seconda `hd0,1`, mentre la prima partizione del disco

fisso slave sarà hd1,0 ecc. Inoltre Grub non differenzia i dischi IDE da quelli SCSI, per cui la sintassi appena indicata funziona con entrambi i tipi di unità. Infine, il nome da assegnare al sistema operativo (e la corrispondente etichetta di menu) si chiama «title», e non «label» come in Lilo, ma anche qui è un nome a piacere. Queste e altre differenze possono produrre un iniziale disorientamento, ma è facile adattarsi e saper configurare presto anche questo file.

È possibile passare a Grub un'ampia serie di comandi e di opzioni all'avvio. Se si vuole editare una voce particolare del menu, la si seleziona e quindi si preme **e**; si seleziona poi la riga da modificare e si preme ancora **e**. A questo punto è possibile apportare le modifiche desiderate. Alla fine si preme **Invio** per fissare le modifiche e **b** per riavviare il boot. Per ottenere il prompt dei comandi bisogna digitare **c** e quindi passare le opzioni desiderate. Infine, per interrompere il processo di boot è sufficiente premere **Esc**. Si possono far partire da riga di comando più sistemi, con una serie di comandi del tipo:

```
grub> root (hd0,1)
grub> kernel /boot/vmlinuz root=/dev/hda2 ro
grub> boot
```

Si può anche chiedere a Grub di identificare la partizione in cui risiede il kernel, col comando: `grub> find /boot/vmlinuz`. Si può infine avviare una partizione DOS o Windows (qui presente in `/dev/hda1`) coi comandi:

```
grub> rootnoverify (hd0,0)
grub> makeactive
grub> chainloader +1 grub> boot
```

Anche Grub consente naturalmente il dual boot con Windows e il multiboot fra più distribuzioni GNU/Linux o BSD. Tuttavia, per avviare Windows e non potendolo fare direttamente, Grub si affida a una tecnica chiamata *chain-loading*, per cui in realtà avvia il boot loader nativo di Windows.

10.3.4 Le voci di Grub

Il file di configurazione di Grub è `/boot/grub/menu.lst`. Diamo conto delle sue principali voci e opzioni.

In Grub `global section` troviamo:

timeout = il tempo, in secondi, dopo il quale, se non si effettua alcuna azione, Grub avvia il sistema di default.

default = il sistema da caricare di default, indicato col numero sequenziale con cui compare nel file (anche qui il primo sistema è indicato dallo 0).

color = i colori del menu di Grub, secondo la sequenza: caratteri, sfondo normale, sfondo evidenziato.

In `Other bootable partition config` troviamo:

title = l'etichetta o nome a piacere con cui identificare il sistema operativo, con, in aggiunta, l'indicazione del disco fisso e della partizione in cui risiede.

rootnoverify = disco e partizione su cui procedere per il boot di un sistema operativo non supportato direttamente (lascia al settore di boot della partizione indicata l'onere del bootstrap del sistema).

makeactive = rende la partizione attiva (essenziale per Windows).

chainloader = passa il compito di bootare a un altro boot loader (in questo caso, quello di Windows).

In `Linux bootable partition config` troviamo:

title = l'etichetta o nome a piacere con cui identificare il sistema operativo.

root = la partizione dove si trova la directory di root.

kernel = il path dell'immagine del kernel, più alcune eventuali opzioni per il suo caricamento (come la modalità grafica `vga` o `framebuffer`).

initrd = il path dell'eventuale immagine del ramdisk.

splashimage = il path dell'eventuale immagine di sfondo, da indicare secondo disco, partizione e percorso.

password = la password (eventualmente criptata) per poter modificare i parametri di avvio al boot.

Scheda n. 21

Un esempio di `/boot/grub/menu.lst`

Ecco un caso di `/boot/grub/menu.lst` che monta esattamente gli stessi dispositivi del file `/etc/lilo.conf` che abbiamo riportato sopra, alla Scheda n. 15. Una semplice lettura ne evidenzia le innegabili analogie e, insieme, le differenze di impostazione.

```
# GRUB configuration file '/boot/grub/menu.lst'.
# generated by 'grubconfig'.
mer 16 mar 2005 17:18:07 CET
#
# Start GRUB global section
timeout 10
default 1
color light-gray/blue black/light-gray
# End GRUB global section
# Other bootable partition config begins
title Windows (on /dev/hda1)
rootnoverify (hd0,0)
makeactive
chainloader +1
# Other bootable partition config ends #
Linux bootable partition config begins
title Slackware (on /dev/hda2)
root (hd0,1)
kernel /boot/vmlinuz root=/dev/hda2 ro vga=773
# Linux bootable partition config ends
# Linux bootable partition config begins
title Slack (on /dev/hdb3)
root (hd1,2)
kernel /boot/vmlinuz root=/dev/hdb3 ro vga=773
# Linux bootable partition config ends
# Linux bootable partition config begins
title Debian (on /dev/hdb4)
root (hd1,3)
kernel /boot/vmlinuz root=/dev/hdb4 ro vga=773
initrd = /boot/initrd.img-2.4.27-1-386
# Linux bootable partition config ends
title Install GRUB to floppy disk (on /dev/fd0)
pause Insert a formatted floppy disk and press enter.
root (hd0,1)
setup (fd0)
pause Press enter to continue
title Install GRUB to Linux partition (on /dev/hda2)
root (hd0,1)
setup (hd0,1)
pause Press enter to continue
```



```
title - For help press 'c', then type: 'help'
root (hd0)
title - For usage examples, type: 'cat /boot/grub/usage.txt'
root (hd0)
```

10.3.5 La password di Grub

Anche Grub consente l'inserimento di una password in chiaro, semplicemente aggiungendo la riga:

```
password = password
```

ma (a differenza di Lilo) è in grado anche di criptarla con l'algoritmo MD5. Per questo, si lancia da shell il comando: `grub-md5-crypt`, quindi si inserisce due volte la password (la seconda per conferma), infine si copia in `/boot/grub/menu.lst` il risultato ottenuto:

```
password -md5 passwordmd5
```

La password così ottenuta può essere inserita dopo i codici iniziali e prima dell'inizio della lista dei sistemi operativi per bloccare l'accesso a tutti i sistemi che riportano tale dicitura, secondo l'esempio che segue:

```
# Start GRUB global section
#timeout 30
color light-gray/blue black/light-gray
password -md5 passwordmd5
# End GRUB global section
```

In alternativa, si può collocare una password diversa per ogni sistema, nella riga successiva a «title». In questo caso, andrà precisata anche la riga «lock», secondo la seguente sintassi:

```
# Linux bootable partition config begins
title Linux (on /dev/hda2)
lock
root (hd0,1)
kernel /boot/vmlinuz
root=/dev/hda2
ro
vga=773
password -md5
passwordmd5
# Linux bootable partition config ends
```

10.3.6 Reinstallazione di Grub

Nel caso si debba reinstallare Grub, la procedura è del tutto analoga a quella, che già abbiamo visto, per la reinstallazione di Lilo, tranne, ovviamente, gli ultimi passaggi. Si dovrà entrare nella shell di Grub col comando `grub` e apparirà il prompt; si inseriranno quindi tre comandi, il primo (`root`) per specificare l'unità dove risiedono i file di configurazione, il secondo (`setup`) per installare Grub dove desiderato,

il terzo (quit) per uscire dal prompt. Ipotizzando di volerlo collocare nell'MBR del (primo) disco fisso (hd0) e di avere i file di configurazione nella seconda partizione dello stesso (hd0,1), i comandi saranno:

```
root@darkstar:~# grub
Probing devices to guess BIOS drives. This may take a long time.

GNU GRUB  version 0.97  (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> root (hd0,1)
  Filesystem type is ext2fs, partition type 0x83

grub> setup (hd0)
  Checking if "/boot/grub/stage1" exists... yes
  Checking if "/boot/grub/stage2" exists... yes
  Checking if "/boot/grub/e2fs_stage1_5" exists... yes
  Running "embed /boot/grub/e2fs_stage1_5 (hd0)"...  18 sectors are embedded.
succeeded
  Running "install /boot/grub/stage1 (hd0) (hd0)1+18 p (hd0,1)
/boot/grub/stage2 /boot/grub/menu.lst"...  succeeded
Done.

grub> quit
```

10.4 Boot manager indipendenti

Se usiamo Lilo o Grub, l'avvio di altre distribuzioni presenti nel disco fisso (nel nostro esempio, Debian) è legato a Slackware, visto che è il Lilo di quest'ultima che gestisce tutti i programmi. Nel caso si cancelli Slackware, o si aggiorni Debian modificandone il kernel, si dovrà riconfigurare Lilo. Un'alternativa che eviti questa scomodità è ricorrere a un boot manager come **xols** (scaricabile all'indirizzo: <http://www.ranish.com/part/xosl.htm>) oppure **GAG** (Graphical Boot Manager), il quale consente multi-boot e continua a risiedere nell'MBR del tutto indipendentemente dai sistemi operativi installati. Se adottiamo GAG, il Lilo di Slackware dovrà essere installato non nell'MBR, ma nel primo settore della partizione di root per poter essere richiamato da GAG. GAG si può scaricare dalla sua homepage, all'indirizzo <http://gag.sourceforge.net/index.html>. Descriviamo velocemente l'installazione di sicurezza di GAG, cioè in un floppy, per verificarne il funzionamento. La procedura si articola nelle tappe seguenti:

- 1) si scompatta il file .zip che contiene GAG
- 2) si copia su floppy il file disk.dsk con il comando: `dd if=disk.dsk of=/dev/fd0 bs=512 count=2880`
- 3) si avvia il sistema dal floppy e si configura GAG scegliendo in sequenza le voci: **Installazione, Tastiera** (le tastiere italiane sono di tipo «qwerty» dalle prime cinque lettere dei tasti collocati in alto a sinistra), **Lingua, Configurazione, Aggiungi un sistema operativo** e infine **Boot da floppy**.

Una volta salvate le impostazioni, si può provare il boot. Se tutto funziona e siamo soddisfatti del nostro gestore di boot, possiamo entrare nuovamente nel menu di configurazione per salvarlo, questa

volta, sull'MBR del disco fisso. Ci sono naturalmente anche altri bootmanager indipendenti, fra cui **U-Boot** (<http://u-boot.sourceforge.net>) e **Smart BootManager** (<http://btmgr.webframe.org>).

Capitolo 11

Gestione dei pacchetti

11.1 I pacchetti .tgz

Il sistema di gestione dei pacchetti di Slackware utilizza degli archivi (in formato .tar) compressi in modo comune (con gzip): la loro estensione è infatti .tgz. Il sistema ha un database che tiene conto dei pacchetti installati, facilitando il loro aggiornamento o la loro rimozione. L'utilità pseudo-grafica che semplifica questo compito è **pkgtool**. È possibile effettuare gli stessi compiti anche con alcuni comandi della console, metodo semplice e più veloce ancora. Essi valgono però solo ed esclusivamente per il formato .tgz. Se si installano dei pacchetti compilandoli direttamente, da sorgente, allora il database non ne ha notizia; in questo caso, si possono rimuovere dalla directory dei sorgenti, se sono stati conservati e se supportano la funzione `make uninstall`. Nel caso che tale funzione non sia supportata, non resta che rimuovere manualmente file per file, cercando di identificarli mediante i messaggi (soprattutto quelli che seguono al comando `make install`) o un log di installazione.

Scheda n. 22

Le funzioni di Pkgtool

Pkgtool è uno strumento di configurazione proprio di Slackware. Esso svolge diverse funzioni, fra cui quella di gestione dei pacchetti, consentendo di installare un pacchetto dalla directory corrente, da altre directory o da floppy; di rimuovere un pacchetto; di visualizzare il contenuto di un pacchetto e le informazioni su di esso (un'analogia utilità di KDE è Kpackage). Infine, alla voce «Setup», consente di gestire parecchi aspetti del sistema, già presentati durante l'installazione. È utile se decidessimo di modificare successivamente le impostazioni, senza volerlo fare modificando manualmente i file di configurazione. Eccone le voci:

05. fontconfig lancia il comando `fc-cache` per indicizzare i font per il server grafico.

70. install-kernel installa un kernel a scelta da un disco di avvio.

80. make-bootdisk crea un floppy di avvio.

90. modem-device consente di selezionare la porta su cui è installato il modem.

hotplug attiva o disattiva hotplug all'avvio.

liloconfig fornisce un'utilità pseudo-grafica per configurare Lilo.

mouse serve per attivare il servizio GPM, cioè il supporto per il mouse nella console.

netconfig configura la rete.

services seleziona i servizi da attivare all'avvio.

setconsolefont cambia i font della shell.

timeconfig seleziona il fuso orario.

xwmconfig seleziona lo windows manager di default.

Esiste inoltre una GUI per X per pkgtool il cui nome è **Xpkgtool**, reperibile all'indirizzo: <http://xpkgtool.sourceforge.net>, in grado di effettuare il controllo dell'integrità dei pacchetti. In più, questa utilità contiene **Slack-Grade** (reperibile però anche separatamente), per il download dei pacchetti.

11.2 Il comando installpkg

Il comando `installpkg` serve a installare un pacchetto. È sufficiente dare, da shell, il comando seguito dal nome del pacchetto da installare: `installpkg nome_pacchetto.tgz`. Esso dispone anche di alcune opzioni; le più utili sono:

-warn: simula soltanto l'installazione e genera un report di quello che accadrebbe installando il pacchetto, report che indirizza allo standard output.

-root: installa il pacchetto in una directory diversa da quella di default, che viene indicata con la seguente sintassi: `installpkg -root /directory`.

Il database dei pacchetti installati risiede in `/var/log/packages`, mentre eventuali script di post-installazione risiedono in `/var/log/scripts/nome_pacchetto`.

Per verificare a quale pacchetto appartenga un determinato file, è in quest'ultima directory che va compiuta la ricerca, col comando:

```
grep [-R] "directory/file" /var/log/packages/* | cut -c 19-
```

Ad esempio, il comando: `grep "bin/bash" /var/log/packages/* | cut -c 19-` produce un output come il seguente:

```
bash-3.0-i486-2:bin/bash2.new
```

da cui si evince il nome del pacchetto, e cioè `bash-3.0-i486-2[.tgz]`.

11.3 Il comando upgradepkg

Il comando `upgradepkg` aggiorna un pacchetto che è già installato con il nuovo pacchetto specificato. Esso effettua dapprima un'installazione di prova, quindi rimuove il pacchetto obsoleto e infine installa effettivamente quello nuovo. È possibile fornire solo il nome del pacchetto nuovo col comando: `upgradepkg nome_nuovo_pacchetto.tgz`, nel caso i due pacchetti abbiano lo stesso nome, ma sia diverso il numero di versione. Se invece essi hanno nomi diversi, è necessario indicarli entrambi; si adatterà pertanto il comando: `upgradepkg nome_vecchio_pacchetto.tgz%nome_nuovo_pacchetto.tgz`, col simbolo della percentuale (detto «carattere di escape») per separare i due nomi.

Fra le sue opzioni, due di una certa utilità sono:

`--install-new`: di norma `upgradepkg` aggiorna semplicemente i pacchetti già presenti, ma non installa pacchetti nuovi. Se, ad esempio, cerchiamo di aggiornare il pacchetto di Abiword che tuttavia non è presente nel sistema, riceveremo un messaggio di errore:

```
bash-3.00# upgradepkg abiword-2.2.8-i486-1.tgz
Error:  there is no installed package named abiword-2.2.8-i486-1.
       (looking for /var/log/packages/abiword-2.2.8-i486-1)
```

Questa opzione, utile se dobbiamo procedere con una serie di più pacchetti alcuni dei quali già presenti in versioni vecchie e altri non installati, consente sia di aggiornare i vecchi sia di installare i nuovi con un unico comando.

`--reinstall`: di norma `upgradepkg` non installa i pacchetti già presenti nella medesima versione. Quest'opzione consente invece al programma di reinstallarli in ogni caso.

11.4 Il comando `removepkg`

Il comando `removepkg` serve a rimuovere il pacchetto con il nome specificato. Pertanto è sufficiente dare il comando: `removepkg nome_pacchetto.tgz`. Esso dispone di alcune opzioni:

- `-warn`: simula soltanto la rimozione e genera un report di quello che accadrebbe rimuovendo il pacchetto, report che indirizza allo standard output.
- `-preserve`: ricostituisce il subtree del pacchetto nella directory `/tmp/preserved_packages/nome_pacchetto.tgz`, dove `nome_pacchetto.tgz` è il nome del pacchetto specificato.
- `-copy`: effettua copia del pacchetto sotto `/tmp/preserved_packages/nome_pacchetto`, ma senza rimuoverlo (ha dunque lo stesso effetto di `-warn -preserve`).
- `-keep`: salva i files temporanei creati da `removepkg`. Utile per propositi di debug.

Il database dei pacchetti rimossi risiede in `/var/log/removed_packages`, mentre quello di eventuali script di post-installazione rimossi risiede in `/var/log/removed_scripts`.

11.5 Il comando `explodepkg`

Il comando `explodepkg` serve semplicemente a scompattare un archivio `.tgz` nella directory corrente.

11.6 Il comando `makepkg`

Il comando `makepkg` serve a generare dei pacchetti `.tgz`.

11.7 La gestione delle dipendenze

I pacchetti `.tgz`, a differenza degli `.rpm`, non gestiscono le dipendenze. In realtà, la gestione delle dipendenze da parte degli `.rpm` si limita a dichiararle, e non anche a soddisfarle. Compilando o installando un `.tgz` dobbiamo invece conoscere in base a quali altri pacchetti esso è stato compilato. Ci sono dipendenze obbligatorie e dipendenze opzionali. Nel primo caso, senza avere installato nel sistema il pacchetto da cui l'altro dipende non saremo in grado di far funzionare l'applicazione. Nel secondo caso

il pacchetto potrebbe comunque non funzionare: lanciandolo da console, esso lamenterà qualche dipendenza insoddisfatta, il che peraltro è un buon sistema per rendersi conto del problema. Se, ad esempio, cerchiamo di avviare una versione di k3b compilata anche col supporto a musicbrainz, che però non abbiamo installato, saremo avvertiti da questo messaggio:

```
samiel@darkstar: $ k3b
k3b: error while loading shared libraries: libmusicbrainz.so.4: cannot open shared
object file: No such file or directory
```

In alternativa, il pacchetto potrebbe essere lanciato, ma senza delle funzionalità originariamente previste.

Questo comportamento è del tutto intenzionale. Slackware, intendendo conferire all'amministratore del sistema il massimo controllo sui pacchetti, lascia che sia l'amministratore stesso, che compila e installa, a decidere manualmente, caso per caso, se e quali dipendenze soddisfare, quali caratteristiche abilitare nei pacchetti e quali invece no, e pertanto rifiuta ostinatamente di ricorrere a strumenti di gestione automatica (come Apt, Yast e lo stesso Swaret).

Si legge infatti in *Slackware Linux Essentials* (a p. 215):

C'è un mito che è circolato da quando RedHat ha introdotto il RedHat Package Manager, e cioè che Slackware non ha alcuna utilità per la gestione dei pacchetti. Ciò è semplicemente falso. Slackware ha sempre incluso una utilità per gestire i pacchetti, anche prima che RedHat esistesse. Benché non ricchissimo di caratteristiche od onnipervasivo come rpm (o, almeno sotto questo aspetto, deb), pkgtool e i programmi correlati sono altrettanto validi per installare pacchetti come gli rpm. La verità su pkgtool non è che non esiste, ma che non effettua alcuna verifica delle dipendenze.

A quanto pare, molta gente nella comunità Linux ritiene che un'utilità per gestire i pacchetti debba includere per definizione la verifica delle dipendenze. Non è così, in quanto almeno Slackware non si comporta in questo modo. Ciò non significa che i pacchetti di Slackware non abbiano dipendenze, ma piuttosto che la sua utilità di gestione dei pacchetti non le verifica. La gestione delle dipendenze è lasciata all'amministratore di sistema, e a noi va bene così.

Alcuni pacchettizzatori integrano comunque il .tgz con un file chiamato «slack-required» nel quale dichiarano le dipendenze, precisandone quando necessario, in aggiunta, le versioni.

Capitolo 12

Creazione di pacchetti .tgz

12.1 La creazione manuale dei pacchetti

Abbiamo detto che i pacchetti di cui fa uso Slackware sono archivi compressi in formato .tgz: si tratta semplicemente di archivi compressi. Se ne guardiamo il contenuto, ci renderemo subito conto che contengono i file da installare in directory identiche a quelle dove andranno collocati nel disco fisso, come se riproducessero al loro interno la destinazione effettiva dei file stessi.

Saper pacchettizzare in .tgz, dal momento che nel mondo dell'open source sono sempre disponibili i sorgenti, significa poter ricorrere a qualsiasi programma nel formato che ci consente di tener conto della sua installazione nel database di Slackware, e dunque di aggiornare o rimuovere il pacchetto stesso in maniera pulita, con `pkgtool` o con i comandi correlati da shell, senza lasciare file abbandonati nel disco fisso (del tutto indipendentemente dal fatto che il pacchetto originale, come spesso non avviene, supporti la funzione «uninstall»), e infine di fruire senza difficoltà di strumenti di aggiornamento automatico come Swaret.

Sebbene esistano degli strumenti per creare i .tgz come **Checkinstall**, la creazione manuale offre indubbi vantaggi: la possibilità di personalizzare il pacchetto per l'architettura del nostro hardware, di abilitare funzioni e plugin che ci servono, di includervi documentazione specifica e altro.

La pacchettizzazione in .tgz è in larga parte meccanica, nel senso che segue alcune regole generali, che funzionano nella maggior parte dei casi. Ciò non toglie che è sempre necessario studiare preliminarmente il pacchetto per capire dove esattamente va installato, con quali opzioni è opportuno configurarlo, se richiede altri pacchetti per poter funzionare (le «dipendenze»). La miglior lettura sul tema altro non è, se non gli SlackBuild originali di Slackware.

Scheda n. 23

Gli strumenti di compilazione

In genere, i pacchetti per GNU/Linux sono open source, per cui ne vengono distribuiti anche i codici sorgente piuttosto che i soli binari già compilati. Questo rende possibile compilarli per architetture (Pentium III, Pentium4, Athlon ecc.) e per sistemi operativi diversi (Fedora e Mandrake usano gli .rpm, Debian i .deb, Slackware i .tgz ecc.). Inoltre consente di abilitare specifiche funzionalità, di modificare i programmi stessi, di correggere bug e così via.

Per compilare un pacchetto da sorgenti sono necessari una serie di strumenti denominati «pacchetti di sviluppo»: tra i principali annoveriamo almeno il compilatore, i file di intestazione, le binutils, il make e perl.

Compendiamo molto sommariamente i passaggi del processo di compilazione: il sorgente (se non è un file unico) contiene una serie di file di estensione `.c` (detti «sorgenti multipli») che C compila trasformandoli in file oggetto di estensione `.o`, un binario quasi completo. Per terminare la compilazione entra così in gioco il linker: può infatti essere necessario combinare più file oggetto per creare un programma e collegare questi ultimi a una o più librerie. Una volta effettuata quest'ultima operazione e posizionati i vari file del pacchetto nelle directory di destinazione, abbiamo pronto il nostro file binario, cioè un eseguibile.

Compilatori

`Gcc` è uno dei compilatori tradizionali: in svariati sistemi Unix, `CC` è un collegamento a `gcc`, e si avvia almeno coi comandi `gcc` e `g++`. Slackware non include l'ormai obsoleto `egcs`. Il compilatore esegue delle operazioni preliminari sul sorgente, prima di passare alla compilazione vera e propria, per renderlo maggiormente comprensibile a `CC`. A tal fine si vale del preprocessore `C` (`cpp`).

I file di intestazione

Detti anche «header», contengono definizioni del sistema operativo di cui il compilatore necessita per svolgere il suo lavoro. Solitamente hanno estensione `.h`. Slackware li presenta nei pacchetti di `glibc`. Inoltre servono gli header del kernel. Di solito, i file di intestazione contengono dichiarazioni di tipi e funzioni che le varie librerie o il sistema operativo offrono.

Le binutils

Contengono una serie di programmi che elaborano e unificano l'output del compilatore per ottenere i binari finali in linguaggio macchina. Si tratta, fra l'altro, dell'assembler `as`, del linker `ld` e del library archiver `ar`. Infatti `CC`, da solo, non è in grado di creare un binario e necessita di determinate librerie, cioè di funzioni precompilate che implementa (se tali librerie sono statiche) oppure che richiama (se esse sono dinamiche).

Le librerie statiche e dinamiche

Le librerie si suddividono in statiche e dinamiche o condivise; le prime hanno estensione `.a`, le seconde `.so`. Quando si collega un programma a una libreria statica, il linker copia il codice dalla libreria all'eseguibile; esso dunque non ha bisogno di trovare altrove, nel sistema, la libreria per essere eseguito, ma più librerie di questo tipo include, più aumentano le sue dimensioni. Per risolvere il problema esistono le librerie condivise. In questo caso il linker si limita a collegare l'eseguibile a una libreria presente altrove nel sistema: si ha così il duplice vantaggio che il programma carica la libreria solo quando ne ha bisogno e non aumenta di dimensione, e che più processi possono condividere lo stesso codice di libreria condivisa in memoria (e su disco). Per verificare quali librerie sta usando un programma, si usa il comando: `ldd nome_programma`.

Il make

`Make` sovrintende al processo di compilazione, per automatizzarne alcuni aspetti. Slackware presenta **GNU make**, che supporta anche alcuni Makefile con estensione non standard. `Make` opera in base a delle regole incorporate, che possono però essere modificate con opportuni parametri. Fra le più significative macro o variabili speciali troviamo **CFLAGS** (opzioni del compilatore C); **LDFLAGS** (come **CFLAGS**, che servono al linker per cercare le librerie); **LDLIBS** (per consentire la ricerca delle librerie condivise in posizioni non canoniche, cioè non incluse nel percorso di ricerca di libreria di runtime preconfigurato); **CC** (il compilatore stesso; l'impostazione predefinita è `cc`); **CPPFLAGS** (opzioni del preprocessore); **CXXFLAGS** (GNU make la usa per i flag di C++; non funziona con altre versioni di `make`). Il `make` rende automatici anche i processi di linking e solitamente è fatto seguire dal comando `make install` per installare effettivamente il programma. Per far sì che il `make` possa adattarsi a varie architetture senza modificare parecchie impostazioni (il che una volta andava effettuato a mano), sono stati introdotti diversi programmi che creano il `make`: fra questi, i più diffusi e funzionali sono lo script **configure** e quindi **autoconf** e **automake**.

Perl

Linguaggio di script richiesto da moltissimi script di configurazione, generatori di codice, parser e utilità correlate.

Altri pacchetti di sviluppo

Vi sono poi ancora altri programmi eventualmente in gioco quando si compila: il debugger (Slackware monta il debugger standard dei sistemi GNU/Linux, e cioè **gdb**), che serve a identificare le cause di eventuali malfunzionamenti (ma serve esclusivamente al programmatore); programmi che leggono file o comandi di configurazione durante la compilazione, come il tokenizzatore **Lex** (in GNU/Linux **Flex**) e l'analizzatore **Yacc** (in Linux, sia **Bison** sia lo stesso **Yacc**).

Linguaggi di scripting

Molti programmi sono oggi redatti con linguaggi di scripting. Dal momento che alcuni di questi hanno raggiunto una grande potenza, non sono pochi i programmatori che li preferiscono a C. Il più importante di questi linguaggi, data la sua estrema versatilità, è, come abbiamo visto, **Perl**; quindi **Python**, **PHP** (per l'elaborazione di ipertesti che si riscontra spesso nelle pagine Web dinamiche e che il più delle volte genera codice HTML); **Java** (poco adottato nei sistemi Unix), e ancora **Tcl**, **m4** (per l'elaborazione delle macro), **Ruby**, **Lisp** (con cui è compilato in buona parte Emacs), **Matlab** (linguaggio commerciale di programmazione matematica a cui attualmente si stanno affiancando progetti open source come **Octave** e **Scilab**).

12.2 Procedura per la creazione di pacchetti .tgz

In linea di massima, la procedura da seguire è la seguente:

1. scompattare il sorgente con `tar`
2. sistemare i permessi dei file con `chmod`
3. esaminare le opzioni da passare al `configure` con `./configure --help`
4. configurare il pacchetto con `configure` e con le opzioni del caso
5. costruire il pacchetto col comando `make`
6. installare il pacchetto col comando `make install`
7. aggiungere la documentazione
8. strappare i binari con `strip`
9. strappare le librerie con `strip`
10. comprimere i file `man` con `gzip`
11. comprimere i file `info` con `gzip`
12. ripristinare i permessi corretti della directory del gruppo `/bin` con `chmod`
13. inserire la descrizione del pacchetto e altri eventuali file informativi
14. creare il pacchetto con `makepkg`
15. cancellare le directory temporanee di lavoro con `rm -r`.

Vediamo ora in dettaglio i vari passaggi.

12.2.1 Scompattare il sorgente

Il sorgente si scompatta da console a seconda del formato, che in genere è `.tar.gz` o `.tar.bz2` (sulla scompattazione e la gestione degli archivi vedi [Scheda n. 24 – La gestione degli archivi](#)), oppure con utilità grafiche come Ark (per KDE) o File Roller (per Gnome) in una directory qualunque. È sempre opportuno evitare di svolgere questa operazione in directory di programma, per non causare eventuali errori di cancellazione, e usare una directory creata *ad hoc*: alcuni la creano in `/home` o all'interno della directory dei sorgenti (soluzioni entrambe scomode), o ancora in una subdirectory di `/tmp`. Qui ipotizzeremo, a titolo di esempio, di lavorare nella directory provvisoria `/tmp/pkg`.

Scheda n. 24

La gestione degli archivi

Vi sono vari tipi di archivi supportati da GNU/Linux, compressi e non. Citiamo qui i principali e indichiamo i più diffusi comandi per la loro gestione.

Il formato .zip

Questo formato non è particolarmente diffuso in ambiente GNU/Linux, e tuttavia è utile conoscerlo per poter gestire gli archivi di ambiente Windows, dove è di fatto lo standard (essendo molto meno diffusi i formati `.rar` e altri).

Zippare un file: `zip nome_archivio.zip nome_file.estensione`. Una volta zippato il file, il programma ci informa sulla percentuale di compressione con l'avviso: «deflated *n%*».

Zippare una directory: `zip [-r] nome_archivio.zip /directory`. L'opzione `-r` consente un processo ricorsivo, cioè esteso alle subdirectory della directory principale. Una volta zippata la directory, il programma ci informa sulla percentuale di compressione dei singoli file.

Szippare un file o un archivio: `unzip nome_archivio.zip`.

Il formato .gzip

Il programma `gzip` è in grado di zippare solo file, e non directory (per compiere questa operazione è necessario combinarlo col programma `tar`).

Comprimere un file: `gzip nome_file.estensione`. L'estensione `.gz` viene aggiunta automaticamente. Questo comando sostituisce il file originario, che viene pertanto eliminato, a meno che non si precisi l'opzione `k`. L'opzione `-1` predispone alla massima velocità dell'operazione, l'opzione `-9` alla massima compressione possibile, passando per i gradi intermedi.

Decomprimere un file: `gzip -d nome_file.zip` oppure: `gunzip nome_file.zip`, dal momento che `gunzip` non è che un link simbolico a `gzip`. Il programma in questione, dopo aver compattato o scompattato, elimina il file originale, a meno che non si precisi l'opzione `k`.

Il formato .bzip2

Il programma `bzip2` è un'altra utilità di sola compressione, ma, sebbene forse un po' più lenta nell'operare e tale da richiedere un uso maggiore di memoria, è in grado di raggiungere percentuali di compressione più elevate di `gzip`.

Comprimere un file: `bzip2 nome_file.estensione`. Questo comando sostituisce il file originario, che viene pertanto eliminato, a meno che non si precisi l'opzione `k`. L'opzione `-1` predispone alla massima velocità dell'operazione, l'opzione `-9` alla massima compressione possibile, passando per i gradi intermedi.

Decomprimere un file: `bzip2 -d nome_file.bz2` oppure: `bunzip2 nome_file.bz2`. Questo comando sostituisce il file originario, che viene pertanto eliminato, a meno che non si precisi l'opzione `k`.

Il formato .tar

Il programma tar serve a creare archivi non compressi. Fra le varie opzioni che supporta servono soprattutto le seguenti:

-f: specifica un file archivio da estrarre o da creare (quest'opzione va sempre specificata, con l'unica eccezione delle unità a nastro).

-c: crea un archivio.

-x: estrae un archivio (è possibile anche estrarre singoli file da un archivio precisando esattamente il loro nome).

-t: mostra il contenuto dell'archivio.

-v: effettua le operazioni in «verbose mode», cioè dando maggiori informazioni.

-p: mantiene le autorizzazioni originali dei file, avendo la precedenza su umask (le autorizzazioni vengono impostate solo al termine del processo di estrazione, per cui è necessario attendere la conclusione dell'intero processo per ottenere da questa opzione il risultato desiderato).

Creare un archivio: `tar -cvf nome_archivio.tar /cartella_da_archiviare.`

Estrarre un archivio: `tar -xvf nome_archivio.tar.`

Listare il contenuto di un archivio (senza estrarre): `tar -tf nome_archivio.tar.`

I formati tar.gz e tar.bz2

Poiché gzip e bzip2 sono solo in grado di comprimere e tar solo di archiviare, è possibile ottenere archivi compressi combinando queste due utilità. Basta aggiungere alle opzioni del comando tar rispettivamente le opzioni z per indicare gzip e j per indicare bz2 al fine di svolgere tutte le operazioni su entrambi i tipi di archivio.

Creare un archivio con gzip (.tgz o .tar.gz): `tar -czvf nome_archivio.tgz /cartella_da_archiviare.`

Creare un archivio con bzip2 (.tar.bz2): `tar -cjvf nome_archivio.tar.bz2 /cartella_da_archiviare.`

Estrarre un archivio con gzip (.tgz o .tar.gz): `tar -xvfz nome_archivio.tgz.`

Estrarre un archivio con bzip2 (.tar.bz2): `tar -xjvf nome_archivio.tar.bz2.` In realtà, l'operazione è duplice: prima i file vengono decompressi, poi estratti. La procedura coincide quindi con l'applicazione di gunzip e quindi di tar -x.

Listare il contenuto di un archivio (senza estrarre): `tar -ztf nome_archivio.tgz.`

Se si estrae un archivio per errore in una qualche directory preesistente, si possono cancellare tutti e solo i file estratti col comando: `tar -ztf nome_archivio.estensione | xargs rm.`

Il formato .rar

L'algoritmo che sta alla base del formato .rar è commerciale. Rar offre un buon fattore di compressione, ma è poco usato in Unix. Per GNU/Linux è stata sviluppata un'utilità a sola riga di comando, reperibile all'indirizzo <http://www.rarlab.com/index.htm>. Essa, se non registrata, consente soltanto di estrarre archivi, con la seguente sintassi: `rar e nome_archivio.rar`. In questo caso si può ricorrere anche a strumenti disponibili in modalità grafica, come Ark).

Il formato .ace

Il formato .ace è commerciale e sviluppato per Windows; presenta un buon equilibrio fra percentuale di compressione e velocità nell'effettuare le sue operazioni. L'utilità per gestire sotto Windows questo tipo di file è inoltre in grado di gestire svariati altri formati di compressione. Esiste un'utilità a riga di comando per GNU/Linux atta a scompattare archivi di questo tipo; si tratta di linunace, reperibile all'indirizzo: <http://www.winace.com/>.

Il formato .7-Zip

7-Zip è un recente formato di compressione sviluppato sotto Windows che riesce a ottenere risultati migliori dei tradizionali algoritmi che stanno alla base di WinZip e di Rar. Ne esiste un

port per GNU/Linux, e più precisamente una utilità da riga di comando reperibile all'indirizzo <http://p7zip.sourceforge.net>. È tuttavia consigliabile usarlo solo per decomprimere file in questo formato, e non anche per creare archivi sotto Unix: esso infatti non mantiene i permessi dei file.

12.2.2 Sistemare i permessi

Per poter lavorare (alcuni rari pacchetti presentano permessi originali non ortodossi, altri pochi ne presentano di strani, come il nome dell'autore), conferiamo a tutti i file il permessi di root (utente e gruppo, cioè root:root) col comando: `chown -R root:root /tmp/pkg`. Quindi sistemiamo i permessi in modo che risultino secondo la tabella seguente:

Permessi originali	Permessi modificati
777	755
775	
771	
611	
555	
666	644
664	
600	
444	
400	

Prima di `chmodare`, verifichiamo naturalmente se ne vale la pena, cioè se effettivamente sono presenti file con quei permessi da modificare. I comandi sono del tipo:

```
find . -perm 775 -exec chmod 755 {} \;
find . -perm 664 -exec chmod 644 {} \;
(che sono i casi più diffusi) e via dicendo.
```

12.2.3 Esaminare le opzioni di configurazione

Il comando `./configure --help` è la fonte primaria di informazioni per esaminare le opzioni da passare al `configure`. Esso precisa di solito in quale directory si installa di default il programma, quali sono le opzioni da passare al compilatore e infine quali sono le opzioni specifiche del pacchetto.

Analizziamo a titolo esemplificativo l'output del comando, passato al sorgente di Rekal 2.1.

Le opzioni di base

Quelle che seguono sono le opzioni di base della configurazione, pressoché identiche per tutti i pacchetti. Forniscono alcune informazioni sul pacchetto, e solitamente nella compilazione si può prescindere da esse.

```
bash-2.05b$ ./configure --help
`configure' configures this package to adapt to many kinds of systems.
Usage: ./configure [OPTION]...[VAR=VALUE]...

To assign environment variables (e.g., CC, CFLAGS...), specify them as
VAR=VALUE. See below for descriptions of some of the useful variables.
```

Defaults for the options are specified in brackets.

Configuration:

```
-h,  --help          display this help and exit
     --help=short    display options specific to this package
     --help=recursive display the short help of all the included packages
-V,  --version       display version information and exit
-q,  --quiet,--silent do not print `checking...` messages
     --cache-file=FILE cache test results in FILE [disabled]
-C,  --config-cache  alias for `--cache-file=config.cache`
-n,  --no-create      do not create output files
     --srcdir=DIR    find the sources in DIR [configure dir or `...`]
```

Le informazioni sull'installazione standard

Quelle che seguono sono le indicazioni sull'installazione standard del pacchetto. Ci suggeriscono che esso andrà a installarsi in `/usr/local` e nelle relative subdirectory.

Installation directories:

```
--prefix=PREFIX      install architecture-independent files in PREFIX
                      [/usr/local]
--exec-prefix=EPREFIX install architecture-dependent files in EPREFIX
                      [PREFIX]
```

By default, ``make install`` will install all the files in ``/usr/local/bin``, ``/usr/local/lib`` etc. You can specify an installation prefix other than ``/usr/local`` using ``--prefix``, for instance ``--prefix=$HOME``.

For better control, use the options below.

Modifica dell'installazione standard

Le generiche indicazioni che seguono suggeriscono come modificare le directory di destinazione di file specifici. In Slackware i file di sistema vanno installati in `/etc`, e per evitare che finiscano sotto `/usr/etc` è necessario passare al configure l'opzione `--sysconfdir=/etc` (sempre che il pacchetto contenga file di questo tipo). Allo stesso modo, file di log ecc. vanno collocati in `/var/log` e non in `/usr/var/log`. A questo fine, si passa al configure l'opzione `--localstatedir=/var`. Infine, i vari Window Manager vanno collocati in `/usr/X11R6`.

Fine tuning of the installation directories:

```
--bindir=DIR          user executables [EPREFIX/bin]
--sbindir=DIR         system admin executables [EPREFIX/sbin]
--libexecdir=DIR      program executables [EPREFIX/libexec]
--datadir=DIR         read-only architecture-independent data [PREFIX/share]
--sysconfdir=DIR      read-only single-machine data [PREFIX/etc]
--sharedstatedir=DIR  modifiable architecture-independent data [PREFIX/com]
--localstatedir=DIR  modifiable single-machine data [PREFIX/var]
```

```

--libdir=DIR          object code libraries [EPREFIX/lib]
--includedir=DIR      C header files [PREFIX/include]
--oldincludedir=DIR  C header files for non-gcc [/usr/include]
--infodir=DIR         info documentation [PREFIX/info]
--mandir=DIR          man documentation [PREFIX/man]

```

Tutte queste attenzioni non servono solo a produrre la standardizzazione dei pacchetti e di conseguenza l'ordine all'interno del disco fisso. Esse fanno anche sì che gran parte del sistema giri in sola lettura, per garantirne la sicurezza.

Prefisso e suffisso

Le indicazioni che seguono sono relative all'assegnazione di un prefisso o di un suffisso al nome del programma al momento dell'installazione. Se non si passa al configure \$ARCH-slackware-linux per specificare il sistema operativo, questo potrebbe generare degli eseguibili con estensione scorretta, come slackware-linux-cups invece, semplicemente, di cups. È proprio per evitare questo genere di errori che si fa ricorso alle opzioni --program-prefix=PREFIX e --program-suffix=SUFFIX.

Program names:

```

--program-prefix=PREFIX      prepend PREFIX to installed program names
--program-suffix=SUFFIX      append SUFFIX to installed program names
--program-transform-name=PROGRAM  run sed PROGRAM on installed program names

```

System types:

```

--build=BUILD          configure for building on BUILD [guessed]
--host=HOST            cross-compile to build programs to run on HOST [BUILD]
--target=TARGET        configure for building compilers for TARGET [HOST]

```

Caratteristiche opzionali del pacchetto

Sono qui di seguito elencate le caratteristiche opzionali del pacchetto. Tra quelle più utili e comuni, quella forse maggiormente adottata è --disable-debug, la quale fa sì che si evitino di ricevere tutti i messaggi di debug (e non di rado produce pacchetti di dimensioni molto più ridotte); del resto questa opzione va abilitata solo dai programmatori. L'opzione --disable-static impedisce la costruzione di librerie statiche, il che è una soluzione adottabile per la maggior parte dei pacchetti, che in tal modo risulteranno anche più snelli.

Optional Features:

```

--disable-FEATURE      do not include FEATURE (same as --enable-FEATURE=no)
--enable-FEATURE[=ARG] include FEATURE [ARG=yes]
--enable-debug         Enables Debug-Code
--disable-dependency-tracking  Speeds up one-time builds
--enable-dependency-tracking  Do not reject slow dependency extractors
--enable-debug=ARG     enables debug symbols (yes|no|full) default=no
--disable-debug        disables debug output and debug symbols default=no
--enable-strict        compiles with strict compiler options (may not work!)

```



```

--enable-warnings      compiles with -Wall and similiar
--enable-profile       creates profiling infos default=no
--enable-pch           enables precompiled header support (currently only KCC) default=no
--enable-final         build size optimized apps (experimental - needs lots of memory)
--disable-closure     don't delay template instantiation
--enable-shared=PKG   build shared libraries default=yes
--enable-static=PKG   build static libraries default=no
--enable-fast-install=PKGS optimize for fast installation default=yes
--disable-libtool-lock avoid locking (might break parallel builds)
--enable-objprelink   prelink apps using objprelink (experimental only tested on linux/386)
--enable-embedded     link to Qt-embedded, don't use X
--enable-qttopia      link to Qt-embedded, link to the Qttopia Environment
--disable-mt           link to non-threaded Qt (deprecated)
--enable-kernel-threas Enable the use of the LinuxThreads port on FreeBSD/i386 only
--disable-threading   disables threading even if libpthread found

```

Personalizzazione del pacchetto

Qui di seguito si dettaglia che il pacchetto può essere installato abilitando determinate funzioni legate ad altri pacchetti presenti nel sistema. Ad esempio, l'opzione `--with-xinerama` fa sì che sia abilitato il supporto per Xinerama. Molti pacchetti consentono di abilitare o meno specifiche utili opzioni, per adattare il pacchetto alle proprie esigenze. Ad esempio, alcune GUI per database consentono di usare vari database (da MySQL a PostgreSQL) a seconda delle opzioni attivate; per certi programmi di posta elettronica, può essere opportuno abilitare `ssl` con l'opzione `--enable-ssl`, per altri determinati plugins col comando `--enable-plugins` e via dicendo. Altre opzioni suggeriscono al programma la presenza di librerie in posizioni non canoniche, che potrebbero sfuggire al controllo automatico. Alcune funzioni sono abilitate se il programma trova determinate librerie già presenti nel sistema, senza il bisogno di precisare esplicitamente dei flag. Così, `k3b` attiva i plugin per decodificare alcuni formati di file sonori semplicemente se individua le relative librerie.

Optional Packages:

```

--with-PACKAGE[=ARG]   use PACKAGE [ARG=yes]
--without-PACKAGE     do not use PACKAGE (same as -with-PACKAGE=no)
--with-gnu-ld          assume the C compiler uses GNU ld default=no
--with-pic             try to use only PIC/non-PIC objects default=use both
--with-pythondir=pythondir use python installed in pythondir
--without-python      Do not compile the Scripiter Plugin
--with-xinerama        enable support for Xinerama
--with-extra-includes=DIR adds non standard include paths
--with-extra-libs=DIR  adds non standard library paths
--with-qt-dir=DIR      where the root of Qt is installed
--with-qt-includes=DIR where the Qt includes are
--with-qt-libraries=DIR where the Qt library is installed

```

Variabili di sistema

Sono qui elencate le variabili di sistema. L'ottimizzazione del pacchetto prevede di adottare sempre CFLAGS, cioè di indicare per quale tipo di ambiente deve lavorare il compilatore. I parametri di ottimizzazione da passare al compilatore, che prendono per l'appunto il nome di FLAGS, dipendono anche dall'architettura per cui vogliamo configurare il pacchetto (come Pentium o Athlon).

```
Some influential environment variables:
```

```
CC          C compiler command
CFLAGS      C compiler flags
LDFLAGS     linker flags, e.g. -L<lib dir> if you have libraries
            in a nonstandard directory <lib dir>
CPPFLAGS    C/C++ preprocessor flags, e.g. -I<include dir> if you have
            headers in a nonstandard directory <include dir>
CPP         C preprocessor
CXX         C++ compiler command
CXXFLAGS    C++ compiler flags
CXXCPP      C++ preprocessor
```

```
Use these variables to override the choices made by `configure' or to help it to find
libraries and programs with nonstandard names/locations
```

Se non viene precisato alcun parametro, gcc cerca di minimizzare il tempo di compilazione e lo spazio richiesto in memoria (chiamato «costo di compilazione»), a scapito dell'efficienza del pacchetto. La situazione cambia se al compilatore vengono passati dei parametri definiti. Quelli classici (attualmente ne esistono di nuovi e più spinti, ma tutti ancora a livello sperimentale) sono: `-O` (equivalente di `-O1`. Attenzione: si tratta della lettera *O*, non del numero zero!) riduce le dimensioni del pacchetto; `-O2`: cerca invece di migliorare la dimensione del compilato; infine `-O3`: migliora ulteriormente la velocità del compilato a scapito delle dimensioni (per cui ne produce un lieve incremento). Per compatibilità, nei pacchetti standard, al momento attuale Slackware adotta `-O2`.

Quindi si precisa l'ottimizzazione per il processore e la CPU. Attualmente i pacchetti ufficiali di Slackware sono costruiti per i486, per cui avremo i seguenti due valori: `-march=i486` e `-mtune=i686` (le versioni precedenti di gcc, presenti fino a Slackware 10.2, adottavano `-mcpu` in luogo di `-mtune`). Pacchetti costruiti per uso personale possono anche avere ottimizzazioni più spinte, ad esempio i FLAGS `-O3` e `-march=i686`. Il parametro `-march` è rilevante per i sistemi che dovranno usare il pacchetto (un pacchetto ottimizzato per i686 non funziona su un i484), mentre `-mtune` solo per il sistema che sta compilando il pacchetto, in quanto ne ottimizza le risorse durante il suo lavoro. In ogni caso, si può trovare l'elenco esaustivo delle opzioni al seguente indirizzo: <http://gcc.gnu.org/onlinedocs>. Un altro elenco più semplice ma comunque ampio si legge all'indirizzo: http://gentoo-wiki.com/Safe_Cflags.

L'eventuale opzione `-g` genera informazioni di debug (dunque è l'opposto di `--disable-debug` o di `--enable-debug=no`).

Altre opzioni sono state analizzate nella [Scheda n. 19 – Gli strumenti di compilazione](#). A titolo di esempio, citiamo il caso del pacchetto `openldap`. Il `configure`, se non opportunamente configurato, cerca il Berkeley DB, ma non trova ciò di cui necessita e cioè la versione corretta di quel pacchetto, e di conseguenza interrompe la compilazione col messaggio:

```
checking for Berkeley DB link (default)... no
checking for Berkeley DB link (-ldb43)... no
checking for Berkeley DB link (-ldb-43)... no
checking for Berkeley DB link (-ldb-4.3)... no
```

```
checking for Berkeley DB link (-ldb-4-3)... no
checking for Berkeley DB link (-ldb42)... no
checking for Berkeley DB link (-ldb-42)... no
checking for Berkeley DB link (-ldb-4.2)... yes
checking for Berkeley DB version match... Berkeley DB version mismatch
header: Sleepycat Software: Berkeley DB 3.3.11: (July 12, 2001)
library: Sleepycat Software: Berkeley DB 4.2.52: (December 3, 2003)
no
configure: error: Berkeley DB version mismatch
```

Per risolvere bisogna suggerire al configure il percorso di tale libreria, inserendo i campi CPPFLAGS e LDFLAGS, come segue:

```
CFLAGS="-O2 -march=i486 -mtune=i686" \
CPPFLAGS="-I/usr/include/db4" \
LDFLAGS="-L/usr/lib" ./configure --prefix=/usr \
                                --sysconfdir=/etc \
                                --localstatedir=/var \
                                --enable-debug=no
```

12.2.4 Configurare il pacchetto

Individuate tutte le opzioni che ci interessano, passiamo effettivamente al configure. In primo luogo le variabili di ambiente, quindi il `prefix`, infine le caratteristiche opzionali. In secondo luogo dobbiamo suggerire al configure la directory dove installare il pacchetto. Come già abbiamo visto, a differenza di altre distribuzioni GNU/Linux, in Slackware i pacchetti aggiuntivi in genere non sono collocati in `/usr/local`, ma in `/usr`. Fanno eccezione i pacchetti che girano sotto KDE, che andranno invece collocati in `/opt/kde`. Per suggerire la directory al configure si usa il comando `--prefix=/usr`, oppure, nel secondo caso, `--prefix=/opt/kde`. Il comando risulterà pertanto qualcosa di simile a:

```
CFLAGS="-O2 -march=i486 -mtune=i686" ./configure --prefix=/usr --disable-debug
```

oppure

```
CFLAGS="-O2 -march=i486 -mtune=i686" ./configure --prefix=/opt/kde \
                                                --disable-debug \
                                                --with-jack \
                                                --with-ladspa \
                                                --with-xinerama \
                                                --with-alsa
```

12.2.5 Compilare il pacchetto

Il pacchetto si costruisce semplicemente col classico `make`, a cui in certi casi non frequentissimi va fatto seguire il `make depend` (ma in genere è la procedura stessa che avverte di ciò).

12.2.6 Installare il pacchetto

Dobbiamo ora installare il pacchetto. Tuttavia noi non lo vogliamo installare effettivamente nel sistema per renderlo operativo, ma effettuarne solo un'installazione provvisoria per costruirlo in modo indipendente. A questo fine dobbiamo suggerire al `make install` di installarlo all'interno di una directory temporanea di lavoro, che abbiamo convenzionalmente chiamato `/pkg` e già creato col comando `mkdir -p /tmp/pkg`. Come sappiamo, la struttura di questo pacchetto deve riprodurre la struttura che poi avrà il pacchetto. In altre parole, se determinati file andranno a collocarsi in `/usr/lib`, adesso dobbiamo fare in modo che vadano a finire in `/pkg/usr/lib` ecc. Il comando da impartire per redirigerli è allora:

```
make install DESTDIR=/dir/dir (ad esempio, DESTDIR=/tmp/pkg)
```

oppure:

```
make install prefix=/dir/dir/dir (ad esempio, prefix=/tmp/pkg/usr).
```

Nei programmi più recenti (i Makefile generati con automake e autoconf) l'opzione `DESTDIR` ha soppiantato l'ormai deprecata `prefix` (che, se adottata, produce errori di installazione). Ciò non toglie che un elevato numero di pacchetti la supporti ancora. È da notare che se si adotta `DESTDIR` va indicata solo la directory principale di lavoro (`/pkg`), mentre se si adotta `prefix` va indicata anche la subdirectory (`/pkg/usr`).

In ogni caso, per verificare quale delle due opzioni adottare è sufficiente, dopo aver eseguito il `make`, dare il comando: `grep -R destdir Makefile` oppure cercare la stringa `destdir` o `dc_destdir` nel Makefile con un editor qualsiasi di testo.

12.2.7 Aggiungere la documentazione

Nella directory principale del sorgente è presente un certo numero di file di documentazione (file come `AUTHORS`, `ChangeLog`, `COPYING`, `INSTALL`, `README`, `THANKS`, `TODO` ecc.). Questa documentazione va copiata nella directory di default della documentazione, e cioè in `/usr/doc/nome_pacchetto-versione`. Al momento, naturalmente, la copieremo nella nostra directory provvisoria, di lavoro, e cioè in `/pkg/usr/doc/nome_pacchetto-versione`, che avremo creato apposta col comando:

```
mkdir -p /tmp/pkg/usr/doc/nome_pacchetto-versione.
```

12.2.8 Strappare i binari

Per ridurre le dimensioni del pacchetto vanno strappati i file eseguibili. Essi possono essere cercati manualmente e quindi (posto che siano in `/usr/bin`) strappati con un comando del tipo:

```
strip --strip-unneeded /usr/lib/nome_file
```

o, in alternativa:

```
strip -g /usr/bin/nome_file
```

oppure ancora l'operazione può essere affidata al comando:

```
find . | xargs file | grep "executable" | grep ELF | cut -f 1 -d : | xargs strip \
--strip-unneeded 2> /dev/null
```

Lo `strip` non solo riduce le dimensioni dei binari (e successivamente delle librerie) rendendo il programma più veloce e riducendo l'uso della memoria, ma contribuisce anche a rendere più sicuri i binari: eliminando i simboli di debug dai file, essi risultano diversi da quello normalmente usati in altre distribuzioni e meno soggetti a un possibile exploit. C'è una sola controindicazione, che però riguarda più che altro i programmatori: in caso di crash, il programma non darà indicazioni utili alla soluzione del problema.

12.2.9 Strappare le librerie

Per lo stesso motivo vanno strippate le librerie (sebbene alcuni sostengano che questa operazione può creare dei problemi a certi pacchetti), o manualmente col comando (posto che siano in /usr/lib):

```
strip --strip-unnneeded /usr/lib/nome_libreria
```

o, in alternativa:

```
strip -g /usr/lib/nome_libreria
```

oppure ancora col comando:

```
find . | xargs file | grep "shared object" | grep ELF | cut -f1 -d : |  
xargs strip --strip-unnneeded 2> /dev/null
```

In tutti questi casi, la redirectione dell'eventuale messaggio di errore (2>) a /dev/null evita di ricevere inutili messaggi se i file da strappare in realtà non esistono. La stessa procedura si adotta anche per strappare eventuali file di tipo current o archive. In questo caso il comando sarà:

```
find . -type f -name '*.a' | xargs file | grep  
archive | cut -f 1 -d : | xargs strip --strip-unnneeded 2> /dev/null
```

12.2.10 Comprimere i file man

Sempre per motivi di spazio, i file man, qualora presenti, vanno compressi con gzip, a cui si può passare anche l'opzione -9 per raggiungere le dimensioni più ridotte possibili. Il programma man le scompatta al volo durante la lettura. Il comando allora sarà: `gzip -9 /work/usr/man/*`, oppure: `gzip -9 /work/usr/man/*/*`, nel non infrequente caso che i file di man siano contenuti in sub-subdirectory.

In certi casi, il programma di installazione tenta di installare i file man in /usr/share. Poiché la collocazione canonica è /usr/man (anche se Slackware crea in questi casi dei link), per redirigere la creazione dei file stessi si può passare al configure l'opzione --mandir=/usr. Se questa non dovesse funzionare, si procede spostando manualmente i file man, prima di comprimerli, nella posizione corretta con: `mv /tmp/pkg/usr/share/man /tmp/pkg/usr/man` e quindi rimuovendo la directory originale con: `rm -r /tmp/pkg/usr/share/man`.

12.2.11 Comprimere i file info

Ancora una volta per ridurre al massimo le dimensioni del pacchetto, si comprimono sempre con gzip i file info, qualora presenti. Il comando allora sarà: `gzip -9 /work/usr/info/*`.

Infine si rimuove il file dir, spesso presente, che, in quanto contiene semplicemente l'elenco dei file del pacchetto, è considerato inutile.

12.2.12 Ripristinare i permessi

Fino a Slackware 10.2, il «pacchetto perfetto alla Slack» voleva che i binari (contenuti in /bin, /usr/bin o /sbin, /usr/sbin e /usr/X11R6/bin) fossero proprietà di root:bin. Questi permessi erano pertanto ripristinati (dal momento che essi erano invece proprietà di root:root) col comando: `chown -R root:bin /pkg/usr/bin`. Facevano eccezione i binari dei pacchetti da collocare nelle subdirectory di /opt (come /opt/kde/bin oppure /opt/openoffice.org2.0/program), che restavano proprietà di root:root.

Da Slackware 11.0 invece questa posizione è stata ritenuta deprecata: tutti gli eseguibili restano infatti proprietà di root:root. Del resto, a parte alcune remote motivazioni storiche relative ai primi sistemi GNU/Linux, non c'è nessuna ragione pratica o di sicurezza per cui essi debbano appartenere a root:bin. Se i permessi, come dovrebbe, sono già di root:root, non serve dunque modificare nulla.

12.2.13 Inserire la descrizione del pacchetto e altri eventuali file informativi

Lo slack-desc

I pacchetti per Slackware contengono un file di solo testo chiamato di default «slack-desc», contenente una breve descrizione del pacchetto stesso. Esso presenta una formattazione standard. Si tratta obbligatoriamente di undici righe che iniziano tutte col nome del pacchetto (senza alcuna modifica, o il file non risulterà leggibile), precedute da un delimitatore del testo che deve andare dal doppio punto che segue il nome del pacchetto all'ultimo carattere della riga più lunga del testo descrittivo. È conveniente che ogni riga della descrizione resti entro gli ottanta caratteri, per evitare problemi con editor impostati.

Riportiamo un esempio di slack.desc:

```
# HOW TO EDIT THIS FILE:
# The "handy ruler" below makes it easier to edit a package description. Line
# up the first '|' above the ':' following the base package name, and the '
# on the right side marks the last column you can put a character in. You must
# make exactly 11 lines for the formatting to be correct. It's also
# customary to leave one space after the ':'.

      |-----handy-ruler-----|
emacs: emacs (GNU Emacs)
emacs:
emacs: Base binaries and support files for the GNU Emacs editor/environment.
emacs: This version supports X.  If you do not have X installed, you will
emacs: also have to install the replacement Emacs binary that does not
emacs: require the X11, Xaw3d, and Xt libraries.
emacs:
emacs: The emacs binary in this package was configured with these options:
emacs:   i486-slackware-linux --prefix=/usr --with-x11 --with-x-toolkit
emacs:   --with-pop
emacs:
```

La parte superiore, intitolata «HOW TO EDIT THIS FILE», è tutta commentata e consiste soltanto in istruzioni per compilare correttamente la descrizione (ciò nondimeno, è buona norma mantenerla sempre invece di rimuoverla). Il file va collocato in una directory creata appositamente e chiamata /install.

Lo slack-required

Se il pacchetto necessita di librerie particolari, cioè ha delle dipendenze che richiedono altri pacchetti inusuali o comunque non presenti in Slackware, ciò dovrebbe essere segnalato in un file (di solo testo) chiamato «slack-required», che andrà posizionato in /install. Di default, si segnala una sola dipendenza per linea, precisando inoltre la versione richiesta, secondo la sintassi:

nome_applicazione condizione versione,

dove «condizione» può essere = (uguale), < (minore), > (maggiore), >= (maggiore o uguale) e <= (minore o uguale). Ad esempio, la stringa:

```
openh323 >= 1.13.4
```

significa che è richiesta la libreria openh323 in versione uguale o maggiore di 1.13.4.

12.2.14 Creare il pacchetto

Ormai tutto è pronto. Non resta che creare il pacchetto col comando:

```
makepkg opzioni nome_pacchetto-versione_pacchetto-architettura-release.tgz
```

Questa sequenza di dati ha una particolare importanza, soprattutto nel caso si debba aggiornare il pacchetto (col comando `upgradepkg`). Prendiamo ad esempio il pacchetto `emacs-21.3-i486-1ms.tgz`. Il primo campo è il nome del programma (`emacs`); il secondo la versione (`21.3`); il terzo l'architettura per cui è stato compilato (`i486`); il quarto la release (che può essere la prima o le successive, se per qualche motivo il pacchetto viene ricompilato con delle modifiche e dunque sarà `1, 2` ecc.) ed eventualmente una sigla che indica il nome di chi ha costruito il pacchetto (nel mio caso, le mie iniziali e dunque `ms`); infine l'estensione del pacchetto stesso (ovviamente, `tgz`).

Il comando sarà dunque: `makepkg -l y -c n emacs-21.3-i486-1ms.tgz`.

L'opzione `-l y` (dove `-l` significa `linkadd` e `y` abbrevia ovviamente «yes») cancella i link simbolici alle librerie. Se infatti esistono dei link simbolici, essi vengono convertiti in uno script generato automaticamente (`/install/doinst.sh`) che li ricostruisce nel momento dell'installazione del pacchetto.

L'opzione `-c n` (dove `c` significa `chown` e `n` abbrevia ovviamente «no») impedisce che vengano modificati i permessi che abbiamo assegnato. In caso contrario, `makepkg` concederà a tutte le directory e a tutti i file i permessi `755` e la loro proprietà a `root:root`, che è quanto non vogliamo, avendo in precedenza disposto altrimenti.

Un'utile opzione è l'opzione `-p`, che abbrevia `--prepend`. Serve a far sì che, se un pacchetto contiene delle librerie condivise, tutti i link simbolici aggiunti allo script `doinst.sh` saranno premessi allo script esistente. Ciò è utile se tali librerie devono essere linkate prima, perché `doinst.sh` deve farne successivamente uso.

Prendiamo ad esempio il pacchetto di `graphviz`. Su di esso si appoggiano altri pacchetti per generare la documentazione, presupponendo però di trovare in `/usr/lib/graphviz` il file `config`. Certo, la creazione di questo file può essere demandata all'utente, ma si può anche fare in modo che sia il programma stesso di installazione a crearlo. A tal fine andrà inserita in `doinst.sh` la riga:

```
( cd /usr/lib/graphviz ; dot -c )
```

Tuttavia, stando al funzionamento ordinario, essa verrebbe anteposta ai link simbolici che `doinst.sh` crea al momento dell'installazione del programma, e non trovandoli già produrrebbe un errore. Perché invece la creazione dei link preceda l'inserimento della riga, va usata per l'appunto l'opzione `-p`, cosicché il comando di creazione del `.tgz` risulterà:

```
makepkg -l y -c n -p $TMP/$NAME-$VERSION-$ARCH-$BUILD.tgz
```

Saremo avvertiti del corretto procedimento verso la fine del processo di compilazione, quando leggeremo a schermo:

```
Updating your ./install/doinst.sh (prepending symlinks)....
```

Alcuni pacchetti (in genere privi di eseguibili, come i file di linguaggio per altri programmi) non accettano le opzioni per il compilatore, dunque non possono essere costruiti per una particolare architettura. In questo caso, in luogo dell'indicazione dell'architettura (come `i486`) andrà inserita l'indicazione `noarch`.

Creato il pacchetto, si possono rimuovere le due directory temporanee di lavoro (quella in cui abbiamo scompattato il sorgente e quella in cui abbiamo costruito il `.tgz`).

Ora siamo pronti per installare il nostro `.tgz`, ma prima è opportuno verificare che il processo di compilazione sia andato a buon fine. La verifica di eventuali errori si può compiere (è il metodo più

veloce) rileggendo i messaggi che sono apparsi sulla shell, e a questo fine si può impostare la cronologia di una shell come Konsole perché visualizzi un numero illimitato di righe (**Impostazioni – Cronologia – Numero di linee – Illimitato**), oppure – soluzione assai più comoda – si può reindirizzare il processo a un file. Infine, si possono vedere soltanto gli errori reindirigendoli a un file. Nel caso si adotti uno SlackBuild, il comando sarà:

```
./nome_pacchetto.SlackBuild --cleanup 2>/home/errori.txt
```

12.3 Casi particolari

Come abbiamo constatato, la maggior parte dei pacchetti sono forniti di uno script, chiamato *configure*, che genera il Makefile. Poiché però un unico Makefile potrebbe non svolgere il suo lavoro sotto piattaforme diverse, sono stati creati degli script che lo generano automaticamente dopo aver analizzato il sistema per adattarlo a quest'ultimo. Il sistema attualmente più diffuso è *autoconf*. Se invece il pacchetto non contempla *autoconf*, in genere farà uso o di un Makefile personalizzabile, o di un *iMakefile*. Ciò non toglie che anche il Makefile classico, a volte, può essere modificato, con tutto quello che di rischioso implica un'azione del genere.

12.4 Lo script *doinst.sh*

Abbiamo visto che lo script */install/doinst.sh* ha il compito di gestire i link presenti nel pacchetto. Esso può svolgere tuttavia anche delle altre funzioni aggiuntive, fondamentali per pacchetti che presentano caratteristiche peculiari. Se, ad esempio, abbiamo a che fare con un programma che gira in un ambiente particolare, dobbiamo creare degli utenti e un gruppo *ad hoc*. È questo, fra gli altri, il caso di *Qmail*, che gira in ambiente *chroot* per motivi di sicurezza. In questo caso *doinst.sh*, usando il linguaggio del *bash* scripting, dovrà fare in modo che il programma crei i gruppi e gli utenti necessari. Il programma di installazione trova lo script e lo esegue.

Presupponendo che in Slackware sia installato e attivo *sendmail*, lo script deve chiuderne tutte le applicazioni attive, rimuovere e quindi ricreare i link, assegnare i permessi corretti, rimuovere per sicurezza gli utenti già eventualmente presenti se *qmail* è già stato installato in una versione precedente, infine ricreare tali utenti e solo dopo procedere all'effettiva installazione. La parte iniziale di uno script del genere sarà dunque:

```
#!/bin/sh
# chiude tutte le applicazioni attive di sendmail
killall sendmail

# rimuove i vecchi link e ne crea di nuovi
( cd usr/lib ; rm -rf sendmail )
( cd usr/lib ; ln -sf /var/qmail/bin/sendmail sendmail )
( cd usr/bin ; rm -rf sendmail )
( cd usr/bin ; ln -sf /var/qmail/bin/sendmail sendmail )
( cd usr/sbin ; mv sendmail sendmail.replaced )
( cd usr/sbin ; ln -sf /var/qmail/bin/sendmail sendmail )

# assegna dei permessi corretti
chmod 622 var/qmail/queue/lock/trigger
# rimuove vecchi utenti e gruppi
```



```

userdel alias
userdel qmaild
userdel qmail1
userdel qmailp
userdel qmailq
userdel qmailr
userdel qmails
groupdel nofiles
groupdel qmail

# crea nuovi utenti e gruppi
groupadd -g 96 nofiles
groupadd -g 97 qmail
useradd -g nofiles -u 91 -d /var/qmail/alias alias
useradd -g nofiles -u 92 -d /var/qmail qmaild
useradd -g nofiles -u 93 -d /var/qmail qmail1
useradd -g nofiles -u 94 -d /var/qmail qmailp
useradd -g qmail -u 95 -d /var/qmail qmailq
useradd -g qmail -u 96 -d /var/qmail qmailr
useradd -g qmail -u 97 -d /var/qmail qmails
.
.
.

```

È possibile anche inserire un avvertimento a video, che metta al corrente, ad esempio, di una libreria necessaria per effettuare l'installazione o di una determinata procedura da seguire. All'interno di `doinst.sh` andranno collocate le righe che si desidera appaiano, con la seguente sintassi:

```

( echo )
( echo )
( echo ===== )
( echo )
( echo Avvertimento )
( echo Avvertimento )
( echo )
( echo ===== )
( echo )

```

12.5 Tips&tricks

12.5.1 Pacchetti con nomi non canonici

Come abbiamo visto, i pacchetti `.tgz` di Slackware presentano dei nomi in forma canonica, composti da una serie di campi (nome, versione, ottimizzazione, numero della compilazione eventualmente preceduto dalla sigla del compilatore), separati da un trattino. Questa forma serve al database per successivi aggiornamenti. Il primo campo è sempre il nome del pacchetto, ed è separato da un trattino dal secondo campo, che è sempre la sua versione, e via dicendo. In certi casi però si trovano dei sorgenti che presentano dei nomi difformi da questa struttura, ad esempio `knights-0.6.4-beta.tar.gz`. Se si compilasse il

.tgz mantenendo questa sequenza, otterremmo un pacchetto dal nome knights-0.6.4-beta-i486-1.tgz: ma allora il programma sarebbe in difficoltà, perché intenderebbe la terza stringa (beta) come un anomalo campo per la compilazione, in luogo di i486 o quant'altro. Di conseguenza, nel .tgz conviene ripristinare l'ordine corretto.

A questo fine si usano (e si riportano nello SlackBuild) due variabili diverse, come \$VERSION e \$SVERSION, in questo modo:

```
VERSION=0.6.4beta
SVERSION=0.6.4-beta.
```

Nello SlackBuild si differenzieranno dove serve, ad esempio:

```
tar xzvf $CWD/$NAME-$SVERSION.tar.gz
.
.
.
mkdir -p $PKG/usr/doc/$NAME-$VERSION
.
.
.
makepkg -l y -c n $TMP/$NAME-$VERSION-$ARCH-$BUILD.tgz
```

12.5.2 Programmi da avviarsi mediante uno script rc

Se il pacchetto creato è tale che il programma dev'essere avviato mediante uno script .rc (di quelli collocati usualmente in /etc/rc.d, oppure come gli window manager che hanno il loro file di avvio xinitrc. in /etc/X11/xinit), allo stesso modo è necessario inserirlo nel pacchetto.

Uno script generico per Window Manager è il seguente:

```
#!/bin/sh
userresources=$HOME/.Xresources
usermodmap=$HOME/.Xmodmap
sysresources=/usr/X11R6/lib/X11/xinit/.Xresources
sysmodmap=/usr/X11R6/lib/X11/xinit/.Xmodmap
if [ -f $sysresources ]; then
    xrdp -merge $sysresources
fi
if
[ -f $sysmodmap ]; then
    xmodmap $sysmodmap
fi
if [ -f $userresources ]; then
    xrdp -merge $userresources
fi
if [ -f $usermodmap ]; then
    xmodmap $usermodmap
fi
# Start the window manager:
exec /usr/X11R6/bin/nome_eseguibile
```

12.5.3 Pacchetti che non compilano tutto

Certi pacchetti complessi non compilano tutto il loro contenuto. Ad esempio, si noterà come Kdepim non compila kmobile. In questo caso una lettura del configure ci porterà a individuare la riga:

```
DO_NOT_COMPILE='$DO_NOT_COMPILE kmobile'
```

In questo modo si ordina al configure di aggiungere kmobile a quanto non dev'essere compilato. Spesso si tratta di utilità o funzioni ancora in testing. Per far compilare anch'esse, basta rimuovere quella riga o, più semplicemente, commentarla. Nel configure si troverà a volte definito in modo esplicito, come nel caso appena considerato, quanto non va compilato, più spesso però saranno presenti delle istruzioni condizionali. Consideriamo un caso esemplificativo, quello di kcardtools. Nel configure troviamo:

```
If test '$FOUND_KSMCARD' != ' YES'; THEN
DO_NOT_COMPILE='$DO_NOT_COMPILE kcardtools'
fi
```

Ciò significa che se la variabile FOUND_KSMCARD ha un contenuto diverso da «yes», allora kcardtools non va compilato. Sempre nel configure, ricercando la stringa FOUND_KSMCARD si trova che la variabile è posta in base a questa condizione:

```
if test $ac_cv_header_kcarddb_h = yes; then
FOUND_KSMCARD='yes'
else
FOUND_KSMCARD='no'
fi
```

Si tratta evidentemente (dal nome della variabile) di un header. Un'ulteriore ricerca sulla stringa ac_cv_header_kcarddb_h ci fa trovare una serie di istruzioni legate al fatto che kcarddb sia stato trovato o meno:

```
if test '$ac_cv_header_kcarddb_h+set' = set; then
echo '$as_me:$LINENO: checking for kcarddb.h'>&5
echo 'ECHO_N checking for kcarddb.h...$ECHO_C'>&6
if test '$ac_cv_header_kcarddb_h+set' = set; then
.
.
.
```

Poiché kcardb.h viene installato insieme a ksmcard (che sta in kdelibs), possiamo concludere che kcardtools viene installato se e solo se trova la presenza di quello header, a sua volta presente solo se abbiamo già installato le kdelibs.

In certi altri casi la presenza di specifici programmi o librerie consente l'attivazione automatica di certe funzioni o supporti o funzionalità aggiuntive. Alcuni programmi riassumono alla fine del configure le opzioni abilitate e quelle che non lo sono. È perciò di solito abbastanza semplice rendersi conto di quello che manca e fare una ricerca sul modo per abilitarlo.

12.6 Creare una patch

Talvolta, per adattare il pacchetto alla nostra architettura o allo stesso sistema operativo, o ancora per risolvere dei problemi di compilazione, è necessario modificare dei file di configurazione originali

(in genere il Makefile, ma talvolta delle librerie). Se si vuole tener traccia delle modifiche (essenziale nel caso poi si intenda produrre uno SlackBuild), si devono modificare i file nella directory di partenza, a cui si aggiunge il suffisso .orig e quindi produrre un file che registri le modifiche apportate col comando: `diff -u -r [file-originale] [file-modificato] >file.diff`.

La patch può essere inserita anche nello SlackBuild, che ricorre a una versione del file .diff compressa con gzip. KDE presenta una comoda e molto efficiente interfaccia per diff e patch, che si chiama **Kompare**. Per apportare modifiche elementari al volo, si usa l'editor **sed**.

Scheda n. 25

Lo streaming editor SED

Sed non è propriamente un editor col quale scrivere dei testi, ma uno «streaming editor», ossia un programma (non interattivo, laddove tutti i tradizionali e ben noti editor, da Vi a Emacs e così via sono interattivi) di manipolazione del testo e più in generale di un flusso di dati, quando esso provenga indifferentemente dallo standard input, da un file o da una pipeline. Il risultato della trasformazione viene restituito attraverso lo standard output (cioè visualizzando il testo modificato a schermo senza intaccare il file originale) o in un file. Ciò risulta particolarmente utile per modificare script di shell, ma anche per produrre il batch editing di file o per creare degli script che modificano in forma automatica dei file esistenti.

Facciamo due esempi banali, immediatamente riproducibili. Nel primo, abbiamo un flusso di testo attivato con echo, in cui si sostituisce al volo una parola con un'altra, visibile sullo schermo:

```
echo 'Il miglior sistema operativo è Windows' | sed /Windows/s//Linux/
```

Nel secondo effettuiamo la sostituzione globale di una stringa all'interno di un file di testo, in modo che il vecchio venga preservato:

```
sed s/'stringa1'/stringa2'/g file_originale.txt > file_modificato.txt
```

La sintassi di sed

Sed opera attraverso un comando, un indirizzo che gli viene fornito e che può essere rappresentato da una stringa di testo, da un numero di riga, da una verifica di occorrenza, da un'espressione regolare. Il testo da elaborare viene racchiuso fra delimitatori, costituiti solitamente da semplici barre (/) – ma anche dal trattino basso (_) oppure dalle virgole, o ancora dalla barra verticale (!): quello che conta è soltanto la leggibilità della formula –, cui si aggiungono determinati operatori.

Ad esempio, `3d` indica a sed di cancellare la terza riga dell'input, mentre `/parola/d` gli ordina di cancellare la parola `parola`.

Sed adotta l'opzione `-e` per indicare che quanto segue è un comando. Se è presente una singola istruzione, allora tale opzione può essere omessa.

Negli script di shell l'adozione degli apici semplici evita l'espansione di shell. Comandi lunghi possono essere scritti su più righe usando la barra inversa (\) come indicatore di a capo.

Possono essere adottate anche le espressioni regolari, come `^` (che indica l'inizio di riga), `$` (che indica la fine di riga), `.` (che indica un singolo carattere qualsiasi), `*` (che indica zero o più ripetizioni del carattere precedente), `[]` (che indica tutti i caratteri all'interno delle graffe). Di conseguenza, avremo fra l'altro che:

<code>/./</code>	restituisce le linee contenenti un solo carattere
<code>/././</code>	restituisce le linee contenenti due soli caratteri
<code>/^#/</code>	restituisce le linee contenenti «#»
<code>/^\$/</code>	restituisce le linee vuote
<code>/a ~/</code>	restituisce le linee che terminano con «a»

<code>/a*/</code>	restituisce le linee che terminano con «a» seguita da zero o più spazi
<code>/[abc]/</code>	restituisce le linee che contengono «a», «b» o «c»
<code>/^[abc]/</code>	restituisce le linee che iniziano con «a», «b» o «c»

Queste espressioni si usano per tutte le operazioni di sostituzione e cancellazione. Ad esempio, il comando: `sed -e '/regexp/d' /percorso/file | more` elimina le righe corrispondenti all'espressione regolare indicata, mentre il comando: `sed -n -e '/regexp/p' /percorso/file | more` stampa le righe che soddisfano l'espressione. Infine, il comando: `sed -n -e '/begin/,/end/p' /percorso/file | more` stampa l'intervallo compreso fra le espressioni indicate. Se la prima espressione non viene trovata, non sarà stampato nulla, perché `sed` non sa da dove iniziare a stampare. Se invece non viene trovata la seconda espressione, verrà stampato tutto il testo che segue alla prima, perché `sed` non sa, quando inizia, se e dove la troverà.

I comandi di sed

Fra i comandi di `sed` a cui maggiormente si fa ricorso possono essere citati:

`p` (print) che visualizza l'indirizzo sullo standard output. Pertanto `indirizzo/p` stampa sullo standard output l'indirizzo specificato, facendo vedere sullo schermo una determinata stringa. Il comando: `sed 'p' file.txt` fa vedere sullo schermo il testo di `file.txt`.

`i` (insert) che inserisce una determinata stringa.

`c` (change) che modifica una determinata riga sostituendola con la nuova riga indicata.

`w` (write) che scrive in un file le modifiche apportate, file che viene indicato col carattere di reindirizzamento `>`.

`s` (substitute) che sostituisce una determinata stringa (ed è probabilmente quello a cui si fa ricorso nella più ampia misura). Pertanto il comando `sed indirizzo/s/stringa1/stringa2/` sostituisce in tutte le righe specificate in indirizzo e solo in esse la prima occorrenza di `stringa1` con `stringa2`. Per sostituire tutte le occorrenze è necessario specificare, in aggiunta, l'operatore `g` (global), come abbiamo constatato nei due esempi sopra riportati.

Il comando `d` (delete) che cancella una determinata stringa. Pertanto `indirizzo/d` lo cancella. Ad esempio, lanciando il comando: `sed -e 'd' /etc/lilo.conf` non leggeremo a schermo nulla: `sed` ha infatti aperto il file, ha letto una riga nel suo pattern buffer, ha eseguito il comando di cancellazione e ha proseguito così per tutte le righe. In assenza di indirizzi, `sed` infatti si applica a tutte le linee del testo. L'adozione della virgoletta singola serve a evitare l'espansione di shell. È naturalmente possibile precisare una determinata riga; così, il comando: `sed -e '1d' /etc/lilo.conf` cancella soltanto la prima riga. Gli intervalli si precisano separati dalla virgola, per cui il comando: `sed -e '1,10d' /etc/lilo.conf` eliminerà le prime dieci righe.

Tutti i comandi possono essere applicati con più opzioni aggiuntive. Anche un comando di sostituzione può essere applicato solo a una riga o a un determinato intervallo di righe. Il comando: `sed -e '1,10s/foo/bar/g' file.txt` sostituirà tutte le occorrenze di `foo` con `bar` nelle righe da 1 a 10 incluse del file `file.txt`, mentre il comando: `sed -e '/^$/,/^END/s/foo/bar/g' file.txt` sostituirà tutte le occorrenze di `foo` con `bar` solo nei blocchi di testo che iniziano con una riga bianca e terminano con la stringa `END`.

La barra inversa fa sì che il comando continui sulla riga successiva; in tal modo il carattere di a capo a fine riga diventa una stringa di sostituzione. Ad esempio:

```
s/^*/\  
/g
```

produce come risultato che tutti gli spazi che si trovano a inizio riga vengono sostituiti con un a capo. In pratica, le indentazioni dei paragrafi si trasformano in altrettante righe vuote.

Istruzioni multiple di editing vanno specificate con l'opzione `-e`; se è presente un'unica istruzione, tale opzione può essere omessa.

Un indirizzo seguito da una o più operazioni può richiedere l'adozione di opportune graffe. Ad esempio, per cancellare solo la prima di una serie di righe vuote, si avrà:

```
/[0-9A-Za-z]/,/^$/
/^$/d
```

Alcuni operatori di sed

Elenchiamo alcuni operatori di sed:

<code>n d</code>	cancella la riga <i>n</i> dell'input
<code>/^\$/d</code>	cancella le righe vuote dell'input
<code>1,/^\$/d</code>	cancella le righe dall'inizio alla prima riga vuota dell'input
<code>/stringa/p</code>	visualizza le righe in cui è presente la stringa specificata
<code>s/stringa1/stringa2/</code>	sostituisce in ogni riga la prima occorrenza di <i>stringa1</i> con <i>stringa2</i>
<code>s/stringa1/stringa2/g</code>	sostituisce in ogni riga tutte le occorrenze di <i>stringa1</i> con <i>stringa2</i>
<code>s/ *\$//</code>	cancella tutti gli spazi che si trovano a fine riga
<code>s/00*/0/g</code>	riduce ogni sequenza consecutiva di zeri a un unico zero
<code>/stringa/d</code>	cancella tutte le righe in cui è presente stringa specificata
<code>/stringa//g</code>	cancella tutte le occorrenze della stringa specificata lasciando inalterato il resto delle righe

L'uso di sed negli SlackBuild

Gli SlackBuild altro non sono se non script di shell, e in effetti uno dei principali utilizzi di sed consiste proprio nella sua capacità di modificare al volo tali script, anche in modo molto sofisticato. Vediamo un possibile, semplice utilizzo di sed in un paio di SlackBuild, quello di vamps e quello di ocrad.

Nel primo caso, poiché questo programma non ha un configure, alcuni parametri di compilazione vanno inseriti direttamente nel Makefile. Sarebbe certo possibile creare una patch, ma, data la semplicità della modifica da apportare, essa può essere attuata al volo appunto mediante sed. Si tratta di suggerire al Makefile di adottare determinate ottimizzazioni inserendo la riga corrispondente con la seguente sintassi: `sed -i "s/-03/-03 -march=i486 -mtune=i686/g" vamps/Makefile`.

Nel secondo caso, le righe:

```
sed -i "s//usr/local//usr/g" configure
./configure
sed -i "s/-Wall -W -O2/-O2 -march=i486 -mtune=i686/g" Makefile
```

passano al configure delle opzioni di configurazione e di ottimizzazione.

12.7 Velocizzare le ricompilazioni

Chi affronta un gran numero di ricompilazioni può velocizzare il processo ricorrendo a ccache, una cache per compilatore che sveltisce la ricompilazione di codice C/C++ memorizzando le precedenti compilazioni e individuando, per ogni nuova compilazione, se essa è già stata compiuta in precedenza. Per attivare questa funzione bisogna creare dei link simbolici dai compilatori a ccache. In concreto, si procederà con i comandi che seguono:

```
cd /usr/local/bin
ln -s /usr/bin/ccache c++
```

```
ln -s /usr/bin/ccache cc
ln -s /usr/bin/ccache g++
ln -s /usr/bin/ccache gcc
ln -s /usr/bin/ccache i486-slackware-linux-c++
ln -s /usr/bin/ccache i486-slackware-linux-g++
ln -s /usr/bin/ccache i486-slackware-linux-gcc
```

I link sono inseriti in `/usr/local/bin` perché esso viene considerato prima di `/usr/bin` e di `/bin`, per cui lanciando `gcc` verrà richiamato `ccache` e non `gcc` stesso.

12.8 Il «fake install»

La procedura descritta funziona per la maggior parte dei pacchetti. Tuttavia a volte si incontrano degli inconvenienti, soprattutto se il pacchetto non contiene un `configure`: alcuni `Makefile` non contemplano la presenza di `DESTDIR` per redirigere a piacimento l'installazione. Per questi esistono vie alternative; è possibile cambiare `--prefix` col comando: `make --prefix=/destinazione`, ma spesso questo comando non viene accettato, oppure in altri casi produce dei guai. È allora possibile editare direttamente il `Makefile` apportandovi le modifiche del caso, e cioè sostituendo i path di installazione, ad esempio cambiando con un editor qualsiasi `/usr/local` in `/usr` (in questo caso potrebbe essere anche possibile inserire o modificare i flags per il compilatore).

In altri casi ci si deve rassegnare a produrre un «fake install», cioè una falsa installazione. Ci sono due alternative. Primo caso: dopo il `make` che genera i vari file del pacchetto, non si lancia il `make install`, ma si creano le directory a mano con `mkdir` e si spostano i file al loro interno con `mv` (se si prevede di installare il pacchetto, è sufficiente copiare i file, perché poi essi verranno sovrascritti dall'installazione finale). Ovviamente, è necessario individuare i file creati o modificati dal `make` (in qualche caso ci aiutano le note presenti nei sorgenti; altrimenti è necessario leggere il `Makefile`). Secondo caso: dopo il `make install`, è possibile individuare i file prodotti e installati col comando `find`, dandogli come parametri le varie directory e cercando i file modificati o creati nel minor lasso di tempo possibile, ad esempio: `find /usr -mmmin 1`. Quindi il pacchetto dev'essere ricreato a mano, andando a prendere uno per uno i vari file e spostandoli nella directory di lavoro, anch'esse create manualmente.

Altri programmi ancora non presentano il `configure`, oppure (è il caso di quelli redatti in linguaggio di scripting come Perl e Python) non hanno il `Makefile`. In questi casi è necessario individuare se esiste un altro file adibito all'installazione, altrimenti non resta che effettuare ancora una volta tutto a mano.

12.9 Compilazione, kernel e librerie

Alcuni pacchetti richiedono, per funzionare, librerie aggiuntive che vanno pertanto installate. A volte, molte di più sono invece le librerie necessarie per compilare il pacchetto stesso.

In genere, i software applicativi normali, tranne rari casi, non risentono del kernel sotto il quale sono stati costruiti. Un comportamento diverso hanno invece i moduli (o drivers) che dipendono dal kernel per il quale sono stati costruiti: gli `Alsa-driver`, ad esempio, devono essere sempre compilati contro il kernel col quale devono funzionare.

Infine, ci sono degli applicativi che fanno riferimento al compilatore usato per crearli. Possiamo qui ricordare gli applicativi in Perl, il cui funzionamento richiede la presenza nel sistema proprio e solo della versione di Perl con cui sono stati compilati. Un aggiornamento di quest'ultima richiede la ricompilazione di tutto il software.

12.10 Lo SlackBuild

12.10.1 Che cos'è uno SlackBuild

Lo SlackBuild non è che uno script di shell che contiene una serie di comandi atti a costruire un pacchetto .tgz partendo dai sorgenti. È sufficiente collocare nella stessa directory lo SlackBuild, lo slack-desc e il file sorgente (più eventuali file di patch) e lanciare lo SlackBuild stesso. Questo può essere avviato (sempre e solo da root) settando i permessi in modo opportuno (di norma 755) e quindi impartendo il comando: `./nome_pacchetto.SlackBuild` se lo script è eseguibile oppure, in ogni caso, col comando: `sh nome_pacchetto.SlackBuild`. L'aggiunta dell'opzione `--cleanup` effettua la cancellazione delle directory temporanee di lavoro (se lo SlackBuild prevede questa opzione). Il processo può essere seguito da shell o da console, e alla fine nella directory predefinita, che noi poniamo essere `/tmp`, troveremo il pacchetto pronto.

La scrittura di uno SlackBuild è utile perché in tal modo i vari passaggi eseguiti nella creazione del pacchetto non vanno persi: sarà sempre possibile compilare una versione aggiornata (controllando i cambiamenti) senza dover rifare daccapo tutto il procedimento. Inoltre, modificando alcuni parametri, si può compilare in modo veloce un pacchetto personalizzato (ad esempio, è possibile utilizzare uno SlackBuild ufficiale e modificare i FLAGS per adattarlo maggiormente alla nostra architettura). Naturalmente, la stesura dello SlackBuild richiede una qualche conoscenza di bash scripting.

12.10.2 Analisi di uno SlackBuild

Ci limitiamo in questa sede a offrire un esempio commentato di SlackBuild, relativo all'editor Nano.

Nella sezione che segue si imposta la shell con cui far funzionare lo SlackBuild, nella fattispecie bash (mediante l'istruzione bang della prima riga). Infatti altre shell usano comandi leggermente diversi. Si fornisce opzionalmente anche l'indicazione del sito da cui è tratto il sorgente e magari il nome di chi ha effettuato la pacchettizzazione. Tutta questa breve sessione è commentata in quanto non fa parte dello script vero e proprio, cioè di quanto verrà in effetti eseguito.

```
#!/bin/sh
#
# Heavily based on the Slackware 10.2 SlackBuild
# http://www.nano-editor.org
```

Dapprima (prima riga) si imposta la directory corrente in cui stanno il sorgente, lo slack-desc e lo SlackBuild. CWD è la «Current Work Directory» e il comando `pwd` effettua il «print working directory», che per noi è la directory di esecuzione dello script.

Quindi si impostano la variabile per la directory di lavoro, qui `/tmp` (seconda e terza riga). In secondo luogo vengono create altre variabili atte a velocizzare il lavoro: il nome della directory di lavoro in `/tmp` (quinta riga), il nome del pacchetto (sesta riga), la versione (settima riga), l'architettura (ottava riga) e infine la release del .tgz (nona riga). L'uso delle variabili è sempre preferibile, perché, se si deve aggiornare lo SlackBuild per costruire una versione successiva, basta cambiare una sola occorrenza.

```
CWD=`pwd`
if ["$TMP" = ""]; then
TMP=/tmp
fi
PKG=$TMP/package-nano
NAME=nano
```



```
VERSION=1.2.4
ARCH=i486
BUILD=1ms
```

Qui di seguito si definisce la posizione in cui scompattare il sorgente, e cioè la directory /tmp:

```
if [ ! -d $TMP ]; then
mkdir -p $TMP
fi
```

Qui di seguito si definisce la posizione in cui creare il pacchetto, sempre in /tmp:

```
if [ ! -d $PKG ]; then
mkdir -p $PKG
fi
```

Produciamo adesso, con la sezione che segue, una semplice stampa a schermo, che ci avverte dell'avvio dello SlackBuild per un determinato pacchetto, fornendone nome e versione:

```
echo "+-----+"
echo "| Start SlackBuild $NAME-$VERSION |"
echo "+-----+"
```

Con le righe che seguono ci si porta nella directory di lavoro e si scompatta il sorgente:

```
cd $TMP
tar xzvf $CWD/$NAME-$VERSION.tar.gz
```

Con quanto segue ci si porta nella directory del sorgente ormai scompattato (prima riga), si cambiano i permessi (seconda riga), si rendono ortodossi eventuali permessi anomali (terza e quarta riga):

```
cd $NAME-$VERSION
chown -R root.root .
find . -perm 775 -exec chmod 755 {} \;
find . -perm 664 -exec chmod 644 {} \;
```

Con le righe seguenti si passano al configure i parametri di configurazione per il compilatore, si suggerisce la directory di installazione e si impostano i parametri opzionali:

```
CFLAGS="-O2 -march=i486 -mtune=i686" ./configure --prefix=/usr \
--disable-debug \
--enable-color \
--enable-multibuffer \
--enable-nanorc
```

Le opzioni possono essere collocate tutte sulla stessa riga o, per maggiore leggibilità, su righe diverse. In questo caso, prima di ogni capoverso il simbolo della barra inversa (\) indica che la riga non è conclusa.

Si passa ora alla compilazione vera e propria del pacchetto con la sezione che segue:

```
make
```

Nelle righe che seguono si fa in modo che i file del pacchetto vengano installati all'interno della directory temporanea di lavoro:

```
make install DESTDIR=$PKG
```

Si crea la directory canonica per inserire la documentazione e vi si copiano i file dal sorgente scompattato, in genere presenti nella sua directory principale:

```
mkdir -p $PKG/usr/doc/$NAME-$VERSION
cp -a \
AUTHORS BUGS COPYING ChangeLog INSTALL NEWS README THANKS TODO UPGRADE \
$PKG/usr/doc/$NAME-$VERSION
```

Si strippano i binari e le librerie, qualora presenti, con le seguenti righe:

```
( cd \ $PKG
find . | xargs file | grep "executable" | grep ELF | cut -f 1 -d : | xargs strip \
--strip-unneeded 2> /dev/null
find . | xargs file | grep "shared object" | grep ELF | cut -f 1 -d : | xargs strip \
--strip-unneeded 2> /dev/null
find . -type f -name "*.a" | xargs file | grep archive | cut -f 1 -d : | xargs strip \
--strip-unneeded 2> /dev/null
)
```

Si comprimono i file man e info, qualora presenti, con le seguenti righe:

```
gzip -9 $PKG/usr/man/**
gzip -9 $PKG/usr/info/*
```

Si conferiscono ora i corretti permessi alla directory /usr/bin, dove stanno gli eseguibili:

```
chown -R root.bin $PKG/usr/bin
```

Si crea adesso la directory install e vi si copia lo slack-desc. In questa directory il programma inserirà poi anche l'eventuale doinst.sh:

```
mkdir -p $PKG/install
cat $CWD/slack-desc > $PKG/install/slack-desc
```

Si passa infine nella directory di lavoro del pacchetto appena creato e si impartisce il comando atto a creare il pacchetto stesso con la sezione che segue:

```
cd $PKG
makepkg -l y -c n $TMP/$NAME-$VERSION-$ARCH-$BUILD.tgz
```

Nel caso lo SlackBuild sia stato lanciato con l'opzione `--cleanup`, la sezione che segue dispone di cancellare le due directory temporanee: quella dove è stato scompattato il sorgente e quella in cui è stato creato il pacchetto, che rimarrà, pronto per l'installazione, nella directory `/tmp`.

```
if [ "$1" = "--cleanup" ]; then
rm -rf \ $TMP/\ $NAME-\ $VERSION
rm -rf \ $PKG
fi
```

12.10.3 Tips&tricks per SlackBuild

Inserire avvertimenti a video

Abbiamo visto che è possibile inserire un avvertimento a video mediante lo script `/install/doinst.sh`. Per modificare automaticamente il `doinst.sh` che crea il programma di installazione, nello SlackBuild aggiungendo le righe che vogliamo, va inserita un sezione come la seguente:

```
cat «eof > $PKG/install/doinst.sh
( echo )
( echo )
( echo ===== )
( echo )
( echo Avvertimento )
( echo Avvertimento )
( echo )
( echo ===== )
( echo )
eof
```

Una volta creato il file `doinst.sh`, nello SlackBuild andrà inserita la riga:

```
cat $CWD/doinst.sh > /$PKG/install/doinst.sh
```

e il testo redatto verrà aggiunto al file.

Strappare automaticamente le pagine man

Per strappare le man page eventualmente presenti, senza preoccuparsi di doverle indicare volta per volta, e per evitare messaggi di errore nel caso non siano presenti, basta aggiungere nello SlackBuild una sezione come la seguente (che, con le opportune modifiche, funziona anche per le pagine info):

```
if [ -d $PKG/opt/kde/man ]; then
gzip -9 $PKG/opt/kde/man/man?/*
fi
```

Interrompere lo SlackBuild in caso di errore

Per interrompere immediatamente la compilazione se c'è un errore si inserisce il piccolo script che segue dopo ogni `make` o `make install` dello SlackBuild:

```
if [ $!= 0 ]; then
exit
fi
```

Si deve tuttavia prestare attenzione al fatto che, a volte, questa procedura estrema (anche se effettuata a mano con **Ctrl+z** durante la compilazione) può lasciare indebitamente modificati i permessi di tutta la directory `/tmp` e delle sue subdirectory, assegnando tutto a `root:root`, con conseguenze sgradevoli per gli utenti. In questo caso i permessi si ripristinano, per ciascun utente alla volta, col comando: `chown -R nome_utente:users *nome_utente*`. Allo stesso modo resteranno in `/tmp` le directory di lavoro, da rimuovere manualmente.

In `/tmp` alcuni file sono di `root:users` (fra di essi, quelli contenuti in `.ICE-unix` e anche il file `.X0-lock`. Modificati proprietario e gruppo in `root:root`, essi potrebbero creare dei problemi all'avvio di X. In questo caso, la soluzione radicale è eliminare tutti i file con: `rm -rf /tmp*` e `rm -rf /tmp/*` e riavviare il sistema.

Applicare una patch

Per applicare una patch, nella directory dove stanno il sorgente, lo `slack-desc` e lo SlackBuild andrà collocato anche il file `.diff` zippato con `gzip`. La riga di comando da inserire nello SlackBuild sarà del tipo:

```
zcat $CWD/nome_file.diff.gz | patch -p1 -f --verbose.
```

L'opzione `--verbose` consente di controllare nella shell il felice esito della patch stessa.

12.11 Creare pacchetti .tgz con Checkinstall

Partendo dai sorgenti, è possibile creare pacchetti in formato `.tgz` anche con una procedura automatica, affidandosi a un programma che li compila facendo tutto da solo: **Checkinstall**. Si tratta di un programma che Slackware non installa di default, ma che è reperibile negli `/extra`, nel CD 3 della distribuzione o scaricabile all'indirizzo: <http://asic-linux.com.mx/~izto/checkinstall> (il pacchetto presente in Slackware è tuttavia opportunamente patchato per risolvere un problema relativo all'assegnazione di una forma standard ai nomi dei pacchetti). Dai sorgenti, si deve dare prima `./configure` con le eventuali opzioni e quindi `make`. A questo punto, invece di `make install`, si dà il comando `checkinstall`. Il programma chiede alcune informazioni, fra cui la directory dove creare il pacchetto, un'eventuale descrizione e l'architettura per il quale crearlo (che dovrebbe corrispondere all'opzione passata in precedenza al `configure`). In breve tempo avremo il nostro `.tgz` pronto per l'installazione.

Checkinstall è uno strumento molto comodo, e tuttavia non è sempre infallibile nella gestione di alcuni componenti dei pacchetti: poiché ha un sistema automatico di individuazione dei file di documentazione, se essi hanno nomi poco usuali vengono tralasciati (come `ACKNOWLEDGEMENTS` o `DOCBUILDING`; può non copiare alcuni file di documentazione contenuti in directory diverse da quella principale; non modifica i permessi della directory `/usr/bin`, assegnandoli a `root:root` anziché `root:bin`; infine non effettua sempre il massimo strip possibile delle librerie.

12.12 Creare pacchetti .tgz con Slacktrack

Per ovviare ad alcuni limiti di **Checkinstall**, è stata creata un'altra utilità atta alla costruzione automatica di pacchetti .tgz, **SlackTrack**, presente negli /extra e la cui pagina web è reperibile all'indirizzo: <http://www.interlude.org.uk/unix/slackware>.

In realtà, non si tratta che di uno script di shell che determina un'interazione fra la compilazione da sorgente e il modulo **installwatch** di Checkinstall, per poi creare e eventualmente installare il pacchetto. Slacktrack richiede la creazione di uno script di bash (sul tipo degli SlackBuild) che si appoggia agli SlackBuild per gestire le varie fasi della compilazione e che prende il nome di TrackBuild. Esso si mostra più funzionale di Checkinstall soprattutto nel caso di pacchetti il cui configure non supporta il target DESTDIR per installare il programma in una cartella separata. Secondo gli standard dell'autore, un esempio di comando tipico per compilare, una volta redatto il TrackBuild, è il seguente: `slacktrack -Qp fetchmail-6.2.5.2-i486-1.tgz ./trackbuild.fetchmail`.

12.13 Usare gli .rpm

In Slackware si possono installare i pacchetti in formato .rpm dopo averli trasformati in .tgz col comando: `rpm2tgz`. In questo caso la sintassi è: `rpm2tgz nome_pacchetto.rpm`, dopo di che non ci saranno problemi a installarlo con `pkgtool` o con `installpkg`. Non va tuttavia dimenticato che, in questo caso, il nuovo pacchetto .tgz porta con sé tutta la serie delle sue dipendenze (esattamente come lo .rpm), che complicheranno fatalmente l'installazione.

Inoltre in Slackware si possono installare direttamente i pacchetti in formato .rpm grazie a `rpm`, un'utilità per la gestione dei pacchetti di quel formato. Essa consente di creare, controllare, installare, aggiornare e cancellare gli .rpm. Per dettagli sulle opzioni, si veda: `man rpm`. In ogni caso, è essenziale usare l'opzione `--nodeps` nell'installazione, per scavalcare il problema delle dipendenze. Il tipico comando per installare un .rpm sarà allora: `rpm --nodeps -ivh nome_pacchetto.rpm`, dove l'opzione `-i` (= install) serve appunto a installare, l'opzione `-v` (= verbose) serve a ricevere dei messaggi dettagliati sul procedere dell'operazione, e infine l'opzione `-h` (= hash) stampa a video delle righe composte da 50 cancelletti, mostrando la percentuale dello stato dell'installazione. L'unico inconveniente è che in questo caso il database dei pacchetti installati non terrà conto degli .rpm (essi vengono invece rilevati dal gestore dei pacchetti di KDE, Kpackage).

Capitolo 13

Aggiornare Slackware

13.1 Regole generali

Attualmente, è molto semplice compiere l'aggiornamento automatico di Slackware via Internet. Esistono infatti delle applicazioni che confrontano i pacchetti installati con quelli presenti in alcuni siti detti «repository» e che propongono ed eventualmente installano tali aggiornamenti. Essi inoltre offrono una gestione, peraltro non sempre infallibile, delle dipendenze. Si tratta di **Swaret** (probabilmente il più noto), e quindi di **Slapt-get** ed **Emerde**. C'è infine anche **SlackCheck** (reperibile all'indirizzo <https://georgi.cybcom.net>), atto ad aggiornare più computer: questa utilità opera da un singolo computer tramite **ssh**, generando uno script di upgrade e una lista dei pacchetti non standard per ogni macchina, recuperando poi i pacchetti dai mirror ufficiali. Poiché è capitato che alcuni pacchetti particolarmente delicati aggiornati con questi sistemi automatici abbiano poi dato dei problemi, va prestata la massima attenzione alla loro selezione ed esclusione. Usualmente, conviene escludere quanto meno il kernel, Lilo, i pacchetti di Xorg. In secondo luogo va ricordato che i programmi attivi (come quelli lanciati dai demoni), devono essere chiusi (o «killati») prima di aggiornarli. Altri pacchetti attivi nel server X (come KDE) vanno aggiornati da console, senza lanciare il server X stesso.

13.2 Swaret

13.2.1 Che cos'è Swaret

Swaret (SlackWARE Tool) si sta rapidamente affermando come lo strumento più diffuso e funzionale, tanto che ne esiste la traduzione in svariati linguaggi, italiano incluso, e anche alcune GUI:

Kswaret (basata sulle QT) per KDE (reperibile agli indirizzi: <http://freshmeat.net/projects/kswaret> e <http://kswaret.sourceforge.net>, anche se al momento il progetto sembra abbandonato)

Gswaret (basata su ncurses), progetto che però al momento sembra abbandonato

Zswaret (basata su Zenith), progetto che però al momento sembra abbandonato

Qtswaret (<http://qtswaret.altervista.org> e <http://qtswaret.sourceforge.net/index.php>).

In ogni caso, è estremamente semplice gestire il programma da riga di comando. Slackware non installa Swaret di default, e anzi il pacchetto, reperibile negli `/extra` fino alla versione 9.1, è stato attualmente eliminato; esso è comunque scaricabile all'indirizzo <http://swaret.sourceforge.net>. Swaret è in grado di gestire le dipendenze e (tramite `ldd`) il «track and fix» delle librerie mancanti; ciò non equivale però sempre a una risoluzione esaustiva delle dipendenze, che in certi casi non sono librerie, ma altri tipi di pacchetti. Inoltre Swaret non compie l'aggiornamento dei pacchetti di Dropline Gnome, che si effet-

tua mediante il dropline-installer. Attualmente è in sviluppo un nuovo ramo (il 2) scritto in Perl (il ramo 1 è semplicemente uno script di bash).

13.2.2 Configurare Swaret

Dopo aver installato Swaret, è necessario configurare il file `/etc/swaret.conf.new` dopo averlo rinominato `/etc/swaret.conf` (il principiante incerto sulla configurazione, per questo e per tutti i file di configurazione, dovrebbe rinominarlo non con `mv` ma con `cp`: in questo modo il file originale rimane e in caso di errori può essere ripristinato).

Dobbiamo controllare con un editor che almeno le seguenti righe, le più importanti fra quelle che definiscono le varie opzioni, risultino come descritto qui sotto (consideriamo la versione 1.6.3test3 di Swaret, attualmente – e da parecchio tempo – l'ultima stabile):

`VERSION=10`: serve a stabilire la versione a cui vogliamo aggiornare Slackware. La versione presente in `/etc/swaret.conf` è ormai obsoleta, la 10. Se nel sistema è installata Slackware 11.0, questa voce va necessariamente impostata a `current`, perché altrimenti Swaret non troverà nessun pacchetto più recente di quelli già presenti.

`LANGUAGE=ENGLISH`: specifica la lingua dei messaggi di Swaret. Va modificata in `ITALIANO`.

Poi vanno inseriti i mirror di Slackware, se vogliamo aggiungerne a quelli già presenti di default, con la seguente sintassi:

```
ROOT=PROTOCOLLO:URL
```

ad esempio:

```
ROOT=ftp://ftp.slackware.at/slackware-$VERSION
```

I protocolli possibili sono:

`file` = recupera i pacchetti da un supporto come un CD-ROM, un DVD, un altro supporto magnetico o anche una condivisione di rete.

`ftp` = recupera i pacchetti da un server ftp.

`http` = recupera i pacchetti da un sito Internet.

`rsync` = recupera i pacchetti attraverso il protocollo rsync.

In questa sezione si possono aggiungere parametri ulteriori:

`PACKAGES= 1 o 0`: effettua o meno la ricerca dei soli pacchetti standard nei mirror di Slackware.

`PATCHES= 1 o 0`: effettua o meno la ricerca delle patches nei mirror di Slackware.

`EXTRA= 1 o 0`: effettua o meno la ricerca dei pacchetti extra nei mirror di Slackware.

Infine vanno elencati i repository che ci interessano, con la sintassi:

```
REPOS_ROOT=NOME_REPOSITORY%PROTOCOLLO:URL
```

ad esempio:

```
REPOS_ROOT=Slacky.it%http://www.slacky.it/download
```

Nella sezione `General Settings` sono presenti le seguenti opzioni:

`RANDOMR=0`: impostato a 1, usa i mirror di Slackware a caso, anziché nell'ordine specificato dall'elenco sottostante.

`USEPKGDESC=0`: impostato a 1, cerca anche i pacchetti nella cui descrizione compaiono le parole chiave specificate dall'utente dopo l'opzione `--ACTION`.

`EXCLUDE`: esclude i pacchetti elencati dall'aggiornamento automatico; alcuni, come il kernel e Lilo, sono presenti di default. Possiamo quindi aggiungerne altri. È anche possibile escludere una serie di pacchetti tranne uno o più fornendo una parola chiave, con la seguente sintassi:

```
EXCLUDE=pacchetto[non_escludere_parola_chiave_1, non_escludere_parola_chiave_2,...]
```


Ad esempio, se vogliamo escludere tutti i pacchetti delle lingue di KDE e di KOffice, tranne quello italiano che andrà invece installato, possiamo scrivere:

```
EXCLUDE=kde-i18n[it] koffice-i18n[it]
```

DEPENDENCY=1: questo settaggio controlla le dipendenze, mentre = 0 lo escluderebbe. Sebbene Swaret non sia sempre infallibile nella gestione delle dipendenze, nella maggior parte dei casi produce buoni risultati.

DSEARCHLIB=1: compie la ricerca delle librerie mancanti.

DSEARCHM=0: impostato a 1 fa sì che Swaret ricorra al programma `slocate` per la ricerca delle librerie mancanti (e richiede un aggiornamento frequente del database interno dei pacchetti, col comando `updatedb -uv`). Impostato a 0 effettua la ricerca con `find`.

MD5CHECK=0: impostato a 1, controlla lo MD5SUM dei pacchetti.

GPGCHECK=0: impostato a 1, controlla la firma digitale dei pacchetti (e presuppone che le firme siano già nel keyring del sistema).

DESC=0: impostato a 1, mostra la descrizione dei pacchetti durante l'installazione.

CACHE-DIR=/var/swaret: imposta la directory di cache.

LOG=1: abilita la scrittura di un file di log.

LOG_FILE=/var/log/swaret: imposta la directory del file di log.

WARNINGS=1: abilita i messaggi di avvertimento a schermo.

INFORMATION=1: abilita la visualizzazione delle informazioni a schermo.

Nella sezione `Network Settings` sono presenti le seguenti opzioni:

NIC=eth0: indica la scheda di rete. Questa voce, abilitata di default, va modificata se si ha una connessione a Internet di altro tipo.

NIC=lo: serve per installare da un CD-ROM (attiva il servizio di loopback)

PROGRESS=1: seleziona il tipo di interfaccia per indicare il download.

TIMEOUT=35: imposta il timeout per `wget` e/o `rsync`.

RETRIES=5: imposta il numero di tentativi per `wget` e/o `rsync`.

PASSIVE_FTP=1: attiva la modalità `ftp` passivo (utile se si ha un firewall).

WGGET_PROXY=http://IP or HOST:PORT: se si vuole il supporto proxy ai protocolli.

HTTP o FTP, la presente riga va decommentata e va impostata la relativa variabile.

RSYNC_PROXY=IP or HOST:PORT: se si vuole il supporto proxy al protocollo `RSYNC`, la presente riga va decommentata e va impostata la relativa variabile.

13.2.3 Aggiornare con Swaret

A questo punto, per avviare il processo, bisogna aggiornare il database di `swaret` col comando `swaret --update`, database che si può poi visualizzare col comando `swaret --list`. Quindi col comando `swaret --upgrade` si identificano e installano i pacchetti. L'ulteriore aggiunta dell'opzione `-a` consente di installare i pacchetti automaticamente, senza dover accettare l'aggiornamento uno per uno. Se si desidera installare, aggiornare o rimuovere solo un determinato pacchetto, è necessario dare rispettivamente i comandi: `swaret --install nome_pacchetto`, oppure `swaret --upgrade nome_pacchetto` o `swaret --remove nome_pacchetto`.

È anche possibile scaricare semplicemente un pacchetto senza installarlo col comando: `swaret --get`. In questo caso il pacchetto verrà collocato nella directory `/var/swaret`. In ogni caso, è nelle subdirectory di questa directory che tutti i pacchetti vengono conservati.

Alcune funzioni di Swaret si attivano da riga di comando, secondo la sintassi:

```
swaret --AZIONE [PAROLA_CHIAVE] [OPZIONE]. Elenchiamo tali comandi:
```

```
swaret --changelog: visualizza il Changelog di Slackware, fonte d'informazione principe per gli aggiornamenti e visualizzabile nel web all'indirizzo: http://www.slackware.com/changelog).
```

swaret --compile *parola_chiave* [-a]: scarica e compila il pacchetto individuato mediante la parola chiave. Si tratta di una funzione sperimentale, che si attiva solo se il programma trova le istruzioni in uno buildscript apposito nella directory `/var/swaret/.cache`.

swaret --config: apre il file `/etc/swaret.conf` per modificare la configurazione.

swaret --dep *nome_pacchetto* [-a]: verifica le dipendenze del pacchetto indicato.

swaret --dep *parola_chiave* [-a]: verifica le dipendenze di tutti i pacchetti individuati mediante la parola chiave.

swaret --extra: cerca negli extra di Swaret.

swaret --faq *parola_chiave*: mostra le Frequently Asked Questions di Swaret.

swaret --get *parola_chiave* [-a | -p | -ap | s | -as | -u | -au]: scarica il pacchetto o i pacchetti, le patch o i sorgenti individuati mediante la parola chiave.

swaret --gpg [-i] *path_della_chiave_GPG*: importa una chiave GPG pubblica.

swaret --help: mostra l'indice complessivo dei comandi e delle opzioni.

swaret --htus *parola_chiave*: mostra come usare Swaret attraverso venticinque esempi (la sigla sta per «How To Use Swaret»).

swaret --install *parola_chiave* [-a | -p | -ap]: installa il pacchetto o la patch individuati mediante la parola chiave.

swaret --list [-p | -s | -i | -u | -n | -np]: elenca tutti i pacchetti, le patch e i sorgenti installati o meno nel sistema.

swaret --list | grep *nome_pacchetto*: elenca i pacchetti che rispondono ai criteri impostati mediante la parola chiave.

swaret --log [-t | -c]: mostra il file di log di Swaret.

swaret --manual: mostra la pagina man di Swaret.

swaret --morehelp: mostra l'indice dei comandi e delle opzioni in forma estesa, specificandole una per una.

swaret --purge [-s | -d # | -ds # | -w # | -ws # | -v]: rimuove i pacchetti e patch sorgenti (con l'aggiunta dell'opzione -s) presenti nella cache di Swaret. Non rimuove invece i pacchetti effettivamente installati (azione che vuole il comando `remove`). L'opzione -w serve a specificare una linea temporale, espressa in settimane, mentre l'opzione -d la esprime in giorni; ad esempio, il comando `swaret --purge -w 4` rimuoverà tutti i file più vecchi di quattro settimane e il comando `swaret --purge -ds 5` rimuoverà tutti i sorgenti più vecchi di cinque giorni. Con l'opzione -v questo comando rimuove tutte le versioni dei pacchetti.

swaret --reinstall *parola_chiave* [-a | -p | -ap]: reinstalla il pacchetto o la patch individuati mediante la parola chiave.

swaret --remove *parola_chiave* [-a]: rimuove il pacchetto o la patch individuati mediante la parola chiave.

swaret --resume *parola_chiave* [-p | -s]: riprende lo scaricamento di pacchetti, patch o sorgenti, individuati mediante la parola chiave, che sia stato interrotto.

swaret --search *parola_chiave* [-p | -s | -i | -u | -n | -np]: ricerca pacchetti, patch o sorgenti disponibili, installati e non mediante la parola chiave.

swaret --set *nome_variabile*: cerca una variabile corrispondente a quelle impostate in `/etc/swaret.conf`.

swaret --show *parola_chiave* [-p | -i]: mostra una descrizione per i pacchetti e le patches individuati mediante una parola chiave.

swaret --update: aggiorna il database dei pacchetti di Swaret cercandoli nei mirror e nei repository impostati.

swaret --upgrade *parola_chiave* [-a]: aggiorna tutti i pacchetti o solo quello/i individuato/i mediante la parola chiave.

swaret --version: mostra la versione corrente di Swaret.

Le seguenti opzioni possono anche essere combinate:
l'opzione -a rende il procedimento automatico.

l'opzione `-i` è relativa ai pacchetti già installati.
 l'opzione `-n` riguarda i pacchetti non presenti nel sistema.
 l'opzione `-p` riguarda le patches anziché i programmi.
 l'opzione `-s` è relativa ai sorgenti.
 l'opzione `-u`: aggiorna i pacchetti.

13.3 Slapt-get

Slapt-get, scaricabile all'indirizzo <http://software.jaos.org>, è un sistema APT-like, cioè simile a quello utilizzato da Debian per l'aggiornamento dei pacchetti. A differenza di Swaret, effettua il controllo delle dipendenze in base agli `slack-required`, che vanno pertanto sempre inseriti in ogni pacchetto (ma che, a dire il vero, non fanno parte degli standard ufficiali di Slackware); questa è comunque una regola tassativa per i siti di Slapt-get. Teoricamente, questo metodo è più affidabile, perché sappiamo che non sempre le dipendenze consistono nelle sole librerie. Slapt-get si utilizza da riga di comando con la sintassi `slapt-get [obiettivi] [opzioni]`, ma ne esiste un'interfaccia grafica chiamata **slapt-Gui**, scaricabile all'indirizzo <http://akinimod.sourceforge.net/slaptgui> e basata su GTK.

Il file di configurazione di Slapt-get è `/etc/slapt-getrc`. In esso vengono specificati:

- alla voce `WORKINGDIR` la directory dove Slapt-get piazza i file con cui lavora (gli elenchi dei file dei vari siti, i pacchetti scaricati ecc.);
- alla voce `EXCLUDE` i pacchetti da non aggiornare automaticamente;
- alla voce `SOURCE` gli indirizzi dei siti a cui attingere i pacchetti: un ampio elenco di mirror per Slapt-get si legge all'indirizzo: <http://www.linuxpackages.net/mirrors.php>.

I comandi di Slapt-get sono estremamente simili a quelli di Swaret; di conseguenza, per aggiornare il database dei pacchetti si darà il comando `slapt-get --update`, per aggiornare tutti i pacchetti `slapt-get --upgrade`, per aggiornare un pacchetto `slapt-get --install nome_pacchetto`. Per aggiornare l'intera distribuzione a un'altra versione (quella prescelta nel file di configurazione) si darà il comando `slapt-get --dist-upgrade`. In ogni caso, l'elenco completo dei comandi e delle opzioni, secondo la sintassi: `slapt-get [OPZIONE/I] [OBIETTIVO]`, si ottiene col comando: `slapt-get --help` o consultando la sua pagina `man`. Ne indichiamo solo i principali, in maniera molto più sommaria di quanto sia stato fatto per Swaret:

Obiettivi:

- `--update`: recupera i dati del pacchetto dai mirror.
- `--upgrade`: aggiorna i pacchetti installati.
- `--dist-upgrade`: aggiorna a una nuova release.
- `--install nome_pacchetto`: installa il pacchetto specificato.
- `--remove nome_pacchetto`: rimuove il pacchetto specificato.
- `--show nome_pacchetto`: visualizza una descrizione del pacchetto specificato.
- `--search parola_chiave`: cerca i pacchetti disponibili in base alla parola chiave.
- `--list`: elenca i pacchetti disponibili.
- `--installed`: elenca i pacchetti installati.
- `--clean`: elimina i pacchetti temporanei.
- `--autoclean`: cancella la cache dei pacchetti inesistenti nel sistema.
- `--version`: stampa informazioni su versione e licenza.

Opzioni:

- `--download-only`: si limita a scaricare i pacchetti invece di installarli o di aggiornarli.
- `--simulate` | `-s`: mostra i pacchetti da installare o aggiornare.
- `--no-prompt` | `-y`: non chiedere conferma durante l'installazione o aggiornamento.
- `--reinstall`: reinstalla il pacchetto.

`--ignore-excludes`: esclude dall'installazione o dall'aggiornamento.
`--no-md5`: non esegue il controllo del md5sum.
`--no-dep`: salta il controllo delle dipendenze.
`--ignore-dep`: ignora gli errori nelle dipendenze.
`--print-uris`: si limita a stampare a schermo le url, non recupera i dati.
`--show-stats | -S`: mostra le statistiche dei download.
`--config path`: specifica collocazioni alternative di `slapt-getrc`.
`--remove-obsolete`: rimuove pacchetti obsoleti (solo dopo un `dist-upgrade`).

13.4 Emerde

Emerde, scaricabile all'indirizzo <http://emerde.freaknet.org>, è un port del sistema Portage, specifico di Gentoo GNU/Linux. Questo progetto è molto giovane, e pertanto si deve intendere come fortemente in fase di sviluppo. Anch'esso gestisce in proprio il controllo delle dipendenze.

13.5 Slackupdate

Un altro strumento di aggiornamento è Slackupdate, che altro non è se non uno script scaricabile all'indirizzo: <http://128.173.184.249/slackupdate>. Si tratta di un progetto in corso di sviluppo, attualmente meno elastico e personalizzabile di quelli ci cui ci siamo occupati sopra.

13.6 Aggiornare la versione precedente alla successiva

Allorché esce una nuova versione di Slackware, ci si pone il quesito se sia più opportuno eliminare la vecchia, magari formattando il disco fisso, oppure sovrascrivere la nuova per non perdere dati e impostazioni. Questa seconda possibilità è praticabile, con alcune attenzioni. La procedura per aggiornare una versione precedente all'ultima ufficiale, la 11.0, si sviluppa nei seguenti punti:

- a) se siamo in possesso dei CD-Rom o del DVD di Slackware 11.0, creiamo una directory per gestire i pacchetti della versione nuova a cui aggiornare, directory che qui ipotizziamo di chiamare `/packages`, col comando: `mkdir packages`.
- b) montiamo il CR-ROM sulla directory `/packages` col comando: `mount /dev/cdrom /packages`.
- c) se invece i pacchetti nuovi sono disponibili in altra forma, ad esempio già presenti nel disco fisso, essi vanno copiati in questa directory. Ciò che non può mancare sono comunque i pacchetti fondamentali, quelli presenti nella directory `/slackware` del CD-ROM o del DVD.
- d) ci logghiamo in modalità `single-user`, perciò col comando: `telinit 1`.
- e) aggiorniamo innanzitutto le librerie condivise `glibc` col comando: `upgradepkg /packages/slackware/a/glibc-solibs-*.tgz`.
- f) aggiorniamo l'utilità `pkgtool` col comando: `upgradepkg /packages/slackware/a/pkgtools-*.tgz`.
- g) aggiorniamo o installiamo il programma `sed` (che dovrebbe comunque essere già presente) col comando: `upgradepkg --install-new /packages/slackware/a/sed-*.tgz`.
- h) aggiorniamo i pacchetti già presenti e installiamo quelli nuovi col comando: `upgradepkg --install-new /packages/slackware/*/*.tgz`.
- i) reinstalliamo `lilo` col comando: `lilo [-v]`. Se facciamo il boot da floppy, creiamo un nuovo floppy di avvio col comando: `makebootdisk` usando il nuovo kernel presente in `/boot`.
- l) sistemiamo i file di configurazione. Nella directory `/etc` sono presenti i vecchi file di configurazione, generalmente di estensione `.conf`; alcuni pacchetti ne creano di nuovi, di estensione `.conf.new`. Per spo-

starli in /etc e copiare in essi il contenuto dei vecchi e quindi per rinominare questi ultimi con estensione .conf.bak si può usare uno script di questo tipo:

```
#!/bin/sh
cd /etc
find . -name "*.new" | while read configfile ; do
  if [ ! "$configfile" = "./rc.d/rc.inet1.conf.new" \
    -a ! "$configfile" = "./group.new" \
    -a ! "$configfile" = "./passwd.new" \
    -a ! "$configfile" = "./shadow.new" ]; then
    cp -a $(echo $configfile | rev | cut -f 2- -d . | rev) \
      $(echo $configfile | rev | cut -f 2- -d . | rev).bak 2> /dev/null
    mv $configfile $(echo $configfile | rev | cut -f 2- -d . | rev)
  fi
done
```

m) se stiamo aggiornare da una versione compresa fra la 10.0 e la 10.2, dobbiamo apportare una modifica al file /etc/X11/xorg.conf trasformando il nome del driver della tastiera da Keyboard a kbd con un editor qualsiasi. Se invece proveniamo da una versione ancora precedente (dotata di Xfree e non di Xorg), sarà necessario riconfigurare il server X con l'utilità xorgconfig.

n) torniamo alla modalità multi-user col comando telinit 3.

o) rimuoviamo i pacchetti obsoleti. Essi possono essere verificati dalla directory /var/log/packages col comando: cd /var/log/packages e quindi rimossi con removepkg.

p) rimuoviamo tutti i pacchetti delle lingue per KDE e per Koffice che non ci interessano, e che verosimilmente saranno stati installati dal comando: upgradepkg --install-new. La soluzione più pratica è spostarsi nella directory /var/log/packages col comando cd /var/log/packages e quindi rimuoverli tutti col comando removepkg kde-i18n* koffice-i18n* per poi reinstallare eventualmente solo quelli che ci interessano con installpkg.

Una particolare attenzione va prestata al fatto che, passando da una versione precedente a Slackware 11.0, il pacchetto **devs** non dev'essere aggiornato mentre udev è attivo. La procedura più sicura per effettuare questo aggiornamento consiste nel rendere inesequibile udev con chmod a-x /etc/ec.d/rc.udev, riavviare il sistema e quindi effettuare l'aggiornamento. Se è impossibile riavviare, si possono invece seguire i seguenti passaggi:

- 1) fermare udev con sh /etc/rc.d/rc.udev stop
- 2) smontare pts con umount /dev/pts
- 3) smontare shm con umount /dev/shm
- 4) installare i nuovi pacchetti di devs con installpkg devs*.tgz (o aggiornarli con upgradepkg)
- 5) rilanciare udev con sh /etc/rc.d/rc.udev start
- 6) rimontare tutto il file system con mount -av

13.7 Aggiornare alla current

Aggiornare Slackware alla current significa installare i pacchetti più recenti rispetto all'ultima versione ufficiale, che di volta in volta sono reperibili nel sito di Slackware. Si tratta naturalmente di pacchetti che vengono modificati di continuo, per seguire lo sviluppo delle singole applicazioni, che possono essere instabili e talvolta produrre dei malfunzionamenti del sistema. Tale sviluppo si può seguire andando a leggere i ChangeLog all'indirizzo: <http://www.slackware.com/changelog>.

Fra i diversi metodi che possono essere adottati, uno è, come abbiamo visto sopra, servirsi di Swaret, e poi aggiornando manualmente i pacchetti esclusi da ogni automatismo. Tuttavia può capitare che l'aggiornamento non sia sempre perfetto.

Un sistema interamente «manuale», ma più sicuro, è scaricare i pacchetti della current (comodissimo in questo caso si rivela `rsync`) e procedere quindi, di directory in directory, all'aggiornamento mediante `upgradepkg`. Poiché tuttavia sono presenti alcune modifiche, come i pacchetti ricompilati con dipendenza diverse (ad esempio, nella current che ha preparato Slackware 11.0 Kmail richiedeva le librerie Cyrus-sasl, che nella 10.2 non erano presenti), alcuni pacchetti nuovi andranno installati con `installpkg` o con `upgradepkg --install-new`. Inoltre si dovranno aggiornare i file di configurazione, come abbiamo visto sopra.

Per ottenere sempre le più aggiornate versioni della current, un metodo assai comodo è fare ricorso a `rsync`. Il protocollo `rsync` serve a effettuare in maniera molto efficiente aggiornamenti remoti: esso trasferisce solo le differenze tra i due set di file, quello presente nel computer locale e quello remoto, e di conseguenza automatizza e accelera fortemente la sincronizzazione su rete.

Ad esempio il comando: `rsync -avz --delete --delete-after --progress \`
`rsync://rsync.slackware.at/slackware/slackware-current /archivio` (dove `/archivio` è la directory locale che ospiterà i file scaricati e `slackware.at` il mirror prescelto) scarica i file nuovi della current da `slackware.at`, una volta terminato di scaricare, cancella i vecchi preesistenti, mostrando inoltre il progresso del download. Infatti l'opzione `-v` (= verbose) fornisce informazioni sul server e altro; l'opzione `-a` (= archive mode) attua uno scaricamento ricorsivo di directory e subdirectory; l'opzione `-z` comprime i file durante il trasferimento accelerandolo; l'opzione `--delete` cancella i file locali che non esistono nel server, ma con l'opzione aggiuntiva `--delete-after` li elimina solo dopo che è stato completato lo scaricamento; l'opzione `--progress` mostra la percentuale scaricata e la velocità del download.

Molte altre funzioni di `rsync` possono essere consultate nella guida, col comando: `rsync --help`.

13.8 Creare un DVD della current

È abbastanza semplice trasformare la directory in cui abbiamo scaricato la current in una serie di CD o in un DVD avviabile. Si può creare uno script da lanciare dalla radice della directory in cui c'è la current, il quale creerà in una directory a piacere (nel nostro caso in `/tmp`) un'immagine ISO per un DVD (magari riscrivibile):

```
#!/bin/sh
mkisofs -o /tmp/slackware-install-dvd.iso \
-R -J -V "Slackware Install" -hide-rr-moved \
-v -d -N -no-emul-boot -boot-load-size 4 -boot-info-table \
-sort isolinux/iso.sort -b isolinux/isolinux.bin -c isolinux/isolinux.boot \
-A "Slackware Install DVD" .
```

Si potrà quindi procedere alla masterizzazione con un frontend come K3B oppure, da console, con: `growisofs -Z /dev/nome_dispositivo=/tmp/slackware-install-dvd.iso -dvd-compat -speed=4`. In tal modo ci si troverà in grado di effettuare un'installazione completa e diretta della current.

Capitolo 14

Connettività e reti

La costruzione e la gestione di una rete è operazione delicata e assai complessa: non è questa la sede per occuparcene. Di conseguenza, dopo alcune nozioni in prevalenza teoriche, prenderemo in esame in forma sintetica alcune configurazioni e alcuni comandi utili essenzialmente alla connettività. Prescinderemo invece del tutto dall'analisi di server http, ftp e di posta elettronica.

14.1 Che cosa sono le reti

14.1.1 Le reti informatiche

Una rete informatica è un insieme di dispositivi o computer (detti *host*) indipendenti, ma connessi fra loro in modo da consentire il flusso dei dati. In base alla loro estensione, le reti si possono distinguere in LAN (Local Area Network), e cioè reti locali (come quelle costruite all'interno di un'azienda, ma anche di uno stesso appartamento), MAN (Metropolitan Area Network), e cioè rete di reti locali estesa a livello urbano, WAN (Wide Area Network), rete di reti locali e metropolitane distanti fra loro, poste magari anche in vari continenti.

In base alla loro topologia, ovvero struttura, le reti si possono distinguere in: reti a stella, i cui nodi periferici sono connessi a un nodo principale centrale; reti ad anello, in cui ogni nodo è unito solo al precedente e al successivo; reti a bus, dove tutti i nodi condividono lo stesso mezzo di comunicazione. Perché la rete comunichi con l'esterno e cioè possa entrare in Internet, serve almeno un nodo che colleghi la rete interna con quella esterna, cioè un gateway. In effetti, Internet può essere concepita come una rete logica di enormi dimensioni e complessità, composta da migliaia di reti pubbliche e private.

14.1.2 I protocolli di comunicazione

I protocolli di comunicazione non sono se non le regole in base a cui le informazioni compiono il loro viaggio in rete. Il flusso di dati viene scomposto in pacchetti, ciascuno dei quali contiene ulteriori informazioni, dette «dati di intestazione» (detti *headers*): esse riguardano la scomposizione e la ricomposizione dell'informazione, il controllo dell'integrità del pacchetto, l'ordinamento dei pacchetti per essere ricomposti in modo corretto. I protocolli sono organizzati gerarchicamente in livelli detti *stack* (letteralmente, «pile»). Qui ci interessa in particolare il protocollo TCP/IP (acronimo di «Transmission Control Protocol» e di «Internet Protocol»), lo standard per la comunicazione in Internet.

14.1.3 Il protocollo TCP/IP

Il protocollo TCP/IP è in realtà costituito da una serie, o pila, di protocolli organizzati gerarchicamente in cinque livelli:

il livello fisico, costituito di norma da schede di rete e cavi, oppure dall'etere se ci si riferisce a dispositivi wireless; suo scopo è la trasmissione di un flusso di dati non strutturati attraverso un collegamento fisico, occupandosi della forma e del voltaggio del segnale.

il livello di collegamento, che permette di ottenere il trasferimento affidabile di dati attraverso il livello fisico. Ogni interfaccia di comunicazione (scheda di rete, scheda wireless) ha un suo codice *hardcored* (il suo MAC address) e il suo indirizzo univoco nella rete (l'indirizzo IP).

il livello di rete, costituito dal protocollo IP. Esso ha il compito di consegnare i pacchetti in base a un cammino detto «instradamento» o «routing», senza preoccuparsi del loro ordinamento, compito che spetta a un protocollo di ordine superiore. Per far questo, si basa sul fatto che ogni tappa dal viaggio, cioè ogni host, ha un suo univoco indirizzo IP. Tale protocollo è disponibile in due versioni: il tradizionale ipv4, attualmente il più diffuso e universalmente supportato, e ipv6, un protocollo ancora sperimentale e in genere non adottato dai provider, il quale servirà ad aumentare ad alcune centinaia di miliardi di numeri IP indirizzabili, che nella versione tradizionale è sì ampio, ma limitato a fronte dell'enorme crescita dei server e dei siti.

il livello di trasporto, che a sua volta è distinto in diversi diversi protocolli. Qui ci interessano in particolare TCP e UDP. Il protocollo TCP (Transmission Control Protocol) serve alla connessione affidabile, che effettua il controllo di flusso in modo da essere certi che i dati vengano consegnati senza errori alla destinazione. Il protocollo UDP (User Datagram Protocol) è invece un protocollo inaffidabile, atto a gestire il traffico che non richiede il controllo del flusso di dati.

il livello di applicazione, consistente nel tipo specifico di servizio di rete (condivisione di file system, posta elettronica, siti internet, ssh), ciascuno dei quali in possesso di un suo distinto protocollo (nfs, samba, pop3 ecc.).

Livello fisico	doppino, fibra ottica, cavo coassiale, codifica Manchester, codifica 4B/5B, WiFi, ...
Livello di collegamento	Ethernet, WiFi, PPP, Token ring, ATM, FDDI, ...
Livello di rete	IPv4, IPv6, ARP, DHCP, ICMP, BGP, OSPF, RIP, IGRP, IGMP, Ipsec, ...
Livello di trasporto	TCP, UDP, SCTP, RTP, RTSP, DCCP, ...
Livello applicazione	HTTP, SMTP, POP3, IMAP, FTP, DNS, SSH, IRC, SNMP, SIP, RTP, RSYNC, Telnet, HSRP, ...

14.1.4 L'indirizzo IP

L'indirizzo IP serve a identificare un host all'interno della rete TCP/IP. Fino alla metà degli anni Novanta, gli indirizzi sono stati attribuiti secondo il modello fornito da IANA (acronimo di Internet Assigned Numbers Authority) e consultabile in Internet all'indirizzo: <http://www.iana.org/assignments/port-numbers>, anche se alle assegnazioni effettive erano preposti alcuni organismi distribuiti secondo macro-aree geografiche. Attualmente, questo compito è svolto dall'Internet Corporation for Assigned Names and Numbers (ICANN). Tutti gli indirizzi dovevano appartenere a una determinata classe di reti (classe A, B o C), da cui il nome «classfull» attribuito a questo schema.

Dal 1993 si è affermato un altro schema di indirizzamento, svincolato dal concetto di classe di reti, da cui il nome di «classless», che adotta la notazione CIDR (acronimo di «Classless Inter-Domain Routing»); esso appare maggiormente elastico e in grado di migliorare le prestazioni dell'instradamento IP grazie a una più efficiente organizzazione delle tabelle di routing. Anche in questo caso vi è uno schema mondiale per la distribuzione degli indirizzi, gestito in Europa dal Network Coordination Center (NIC)

del Reseaux IP Europeens (RIPE), sebbene si scontrino in questi caso due diversi criteri: allocazione su base di fornitore o allocazione su base geografica.

Lo schema di indirizzamento classfull

In base al protocollo ipv4, ogni indirizzo IP è composto da 32 bit, suddivisi in quattro gruppi da 8 bit ciascuno. La prima parte serve a identificare la rete, la seconda parte l'host all'interno della rete.

Poiché vi sono reti di dimensioni diverse, gli indirizzi IP sono classificati in cinque classi, indicate con le lettere da A a E.

La classe A di reti ha gli indirizzi col primo bit impostato a 0 e la sua maschera di rete è 255.0.0.0. Ne fanno parte gli indirizzi che vanno da 1.0.0.0 a 127.255.255.255, che pertanto individuano 128 reti da 16.777.214 indirizzi ciascuna.

La classe B di reti ha gli indirizzi col primo bit impostato a 0 e il secondo bit impostato a 1 e la sua maschera di rete è 255.255.0.0. Ne fanno parte gli indirizzi che vanno da 128.0.0.0 a 191.255.255.255, che pertanto individuano 16.384 reti da 65.534 indirizzi ciascuna.

La classe C di reti ha gli indirizzi col primo e il secondo bit impostati a 1 e il terzo bit impostato a 0 e la sua maschera di rete è 255.255.255.0. Ne fanno parte gli indirizzi che vanno da 192.0.0.0 a 223.255.255.255, che pertanto individuano 2.097.152 reti da 256 indirizzi ciascuna.

La classe D di reti ha gli indirizzi coi primi tre bit impostati a 1 e il quarto bit impostato a 0. Ne fanno parte gli indirizzi che vanno da 224.x.x.x a 239.x.x.x. La classe E di reti ha gli indirizzi coi primi quattro bit impostati a 1 e il quinto bit impostato a 0. Ne fanno parte gli indirizzi che vanno da 240.x.x.x a 247.x.x.x. Tali ultime due classi vengono utilizzate per scopi particolari. Gli indirizzi di classe D non identificano né la rete né l'host, bensì un indirizzo multicast: sono infatti utilizzati per le trasmissioni in multicast, dove un solo host trasmette e alcuni altri (quelli facenti parte del gruppo di destinazione) ricevono. Gli indirizzi di classe E sono invece destinati a un utilizzo futuro.

Valori	Uso
da 1.0.0.0 a 9.255.255.255	Blocchi di classe A
10.*.*	Reti private interne
da 11.0.0.0 a 126.255.255.255	Blocchi di classe A
127.*.*	Loopback
da 128.0.0.0 a 172.15.255.255	Blocchi di classe B
da 172.16.0.0 a 172.31.255.255	Reti private interne
da 172.32.0.0 a 191.255.255.255	Blocchi di classe B
da 192.0.0.0 a 192.0.255.255	Riservati
da 192.1.0.0 a 192.167.255.255	Blocchi di classe C
da 192.168.0.0 a 192.168.255.255	Reti private interne
da 192.169.0.0 a 223.255.255.255	Blocchi di classe C
da 224.0.0.0 a 255.255.255.255	Riservati per classi speciali

Questi numeri non sono affatto casuali. L'indirizzo IP si può vedere come una sequenza binaria a 32 bit, esprimibile in notazione decimale come composta da quattro ottetti, ciascuno dei quali a sua volta può assumere un valore fra 0 e 255. Negli indirizzi delle reti di classe A, per definire la segmentazione possibile in reti si utilizza il primo ottetto, il cui valore è compreso fra 1 e 126. Dal momento che il primo bit è impostato a 0, è infatti possibile utilizzare solo i 7 successivi. Abbiamo così 2^7 valori possibili, e cioè 128. Ma poiché il valore 0 deve esprimere la rete locale e il valore 127 identifica l'interfaccia di loopback, restano 126 identificativi che possono essere attribuiti alla rete. Gli altri tre ottetti identificano gli host possibili, e cioè 16.777.214 indirizzi o host per ogni sottorete. Del tutto analogo, *mutatis mutandis*, il calcolo per gli altri tipi di rete.

La scelta della rete da costruire obbedisce a ragioni di opportunità facilmente comprensibili: una grande azienda opererà per una rete di classe B, in modo da attribuire una specifica sottorete a ogni sua divisione, mentre una rete domestica formata da pochi computer troverà la sua conformazione ideale in una rete di classe C.

Il subnetting

Lo schema di indirizzamento classfull è ormai obsoleto e scarsamente rispondente alle esigenze conseguenti all'impetuosa crescita di Internet. Si è infatti affermata la necessità, per molti utenti della rete, di disporre di un numero di indirizzi IP da assegnare alle interfacce di rete superiore a quello gestibile con una rete di classe C (che sono solo 254), ma nettamente inferiore a quello possibile in una rete di classe B (ben 65534). Di consueto, tali utenti si facevano assegnare una rete di classe B, con la conseguenza che molti degli indirizzi possibili rimanevano inutilizzati e in più diminuiva la disponibilità di reti di questo tipo. Inoltre, le tabelle di instradamento dei router delle cosiddette *backbones* (le linee principali di Internet), contenendo un numero enorme di indirizzi, erano diventate ipertrofiche e difficili da mantenere.

Una prima soluzione a questo problema è rappresentata dal subnetting. Il subnetting implica che lo spazio di indirizzamento associato a un indirizzo di una certa classe (indipendentemente dal tipo A, B o C) sia ulteriormente suddiviso in un certo numero di sottoreti. Però, per poter conoscere quale sia la porzione di netid e quella di hostid, è necessario usare maschere di rete più lunghe di quelle associate ai tradizionali indirizzi di classe A, B o C. Inoltre, all'interno di una rete le netmask devono sempre essere indicate esplicitamente, mentre è superfluo all'esterno delle reti si cui è stato praticato il subnetting: infatti, esso influenza il modo in cui è usato lo spazio all'interno di una rete, ma lascia inalterato il modo in cui la rete nel suo insieme viene vista dall'esterno.

Ciò comporta una notevole economia perché, assegnato a un utente un solo indirizzo di classe B, vi si possono creare 128 sottoreti diverse, ciascuna con la capacità di ospitare 510 host. Naturalmente, si potrebbero creare sottoreti in numero minore, ciascuna però con un maggior numero di host.

Lo schema di indirizzamento classless

Lo schema classfull e il subnetting continuano a fare comunque riferimento al concetto di classe di reti. Lo schema classless invece prescinde completamente da questo tipo di approccio e appare pertanto la più efficiente soluzione temporanea ai limiti dei metodi tradizionali, in attesa che si affermi la nuova versione dei protocolli, e cioè ipv6. Tale schema, come abbiamo detto, adotta la notazione CIDR.

Il CIDR elimina completamente la nozione di classe di indirizzamento. Lo spazio di indirizzamento viene semplicemente suddiviso in maniera gerarchica in blocchi contigui di indirizzi, la cui ampiezza è determinata dall'ampiezza della netmask. È così possibile creare non solo delle sottoreti – come abbiamo visto sopra –, ma anche delle super-reti (si tratta del cosiddetto «supernetting») costituite dall'aggregazione di più reti IP. Ad esempio, per un ente che necessita di circa 2000 indirizzi IP una rete di classe B sarebbe troppo grande, per cui è più opportuno assegnargli 8 reti di classe C ($8 \times 256 = 2048$ indirizzi). Supponiamo si tratti delle reti dalla 194.24.0.0 alla 194.24.7.0. Queste 8 reti di classe C possono essere accorpate in un'unica super-rete con le seguenti caratteristiche:

```
Identificativo: 194.24.0.0/21
Supernet mask: 255.255.248.0
Indirizzi: 194.24.0.1 - 194.24.7.254
Broadcast: 194.24.7.255
```

Naturalmente, si richiede un adeguato supporto da parte dei protocolli di routing. L'ultima versione del protocollo BGP, detta BGP-4, supporta il concetto di «supernetting». Per la stessa ragione, sono state apportate delle modifiche anche al firmware dei router.

Poiché spariscono le netmask implicite, ogni indirizzo andrà accompagnato sempre e in maniera esplicita dall'indicazione della netmask. Quest'ultima non viene più indicata con la cosiddetta notazione «dotted decimal» (come 255.255.255.0), ma con la notazione «slash», postposta all'indirizzo IP (come /23), che rappresenta l'ampiezza della stessa espressa nel numero dei bit impostati a 1. In questo modo è possibile, ad esempio, creare una rete con 510 indirizzi (29-2), semplicemente scegliendo il prefisso 23. Contemporaneamente si riducono le routing table, perché le reti che prima erano formate da più reti di classe C, possono essere realizzate con un'unica rete con prefisso minore di 24, in modo da utilizzare un solo riferimento (un unico record) nelle routing table.

Il CIDR – in ciò consiste la sua novità – utilizza maschere di rete di lunghezza arbitraria, mentre l'indirizzamento secondo classi ammetteva tre sole lunghezze della maschera di sottorete (e cioè /8, /16 e /24). Sebbene la maschera di rete di classe C (/27) sia ancora molto utilizzata, attualmente si adottano anche maschere più corte per reti grandi (/23 o /22) e più lunghe per reti piccole (da /25 fino a /39 per reti punto-punto). Con questa netmask si indica quali dei bit che la compongono vanno considerati come indicativi del *net id* e quali dell'*host id*, cioè quale segmento indichi la rete e quale gli host. La notazione usata per esprimere indirizzi CIDR è la seguente: a.b.c.d/x, dove x è il net-prefix, cioè il numero di bit (contati partendo da sinistra) che compongono la parte di indirizzo della rete. I rimanenti bit (32-x) indicano gli host. Il principale vantaggio di questo meccanismo rispetto al precedente consiste nella maggiore flessibilità: con esso non si è obbligati a dedicare alla parte di rete multipli di 8 bit, ma semplicemente il numero di bit più adatto alla situazione che dobbiamo gestire. Ad esempio, se vogliamo che una rete sia costituita dal blocco di indirizzi che va da 220.16.128.0 a 220.16.255.255, allora la notazione CIDR che la rappresenta sarà: 220.16.128.0/17. Allo stesso modo, l'indirizzo IP 192.168.58.147 con netmask 255.255.240.0 può essere scritto 192.168.58.147/255.255.240.0, ma anche, con la nuova notazione, 192.168.58.147/20, mentre l'indirizzo tradizionale 192.168.0.0/255.255.255.0 risulterà 192.168.0.0/24. In pratica, gli indirizzi IP sono scritti postponendo all'indirizzo vero e proprio o la relativa subnet mask (come nel primo caso), oppure il relativo net prefix (come nel secondo caso).

14.1.5 Indirizzi speciali

Ogni nodo o host presenta una fittizia interfaccia di rete che serve a identificare il nodo stesso; a essa è associato un indirizzo. In tal modo, dispositivi o processi che devono fare riferimento al nodo possono farlo senza interferire col funzionamento della rete effettiva. Non a caso, il nodo che ha per indirizzo 127.0.0.1 non è visibile da Internet e definisce il localhost (se nel sistema è stato installato il server web Apache, esso si aprirà a questo indirizzo: basta digitare nel browser `http://localhost` oppure `http://127.0.0.1`). Inoltre, ci sono degli intervalli di indirizzi che non vengono assegnati su Internet. Sono quelli riservati alle reti private interne, le cosiddette Intranet. Per questo uso sono previsti gli indirizzi da 10.0.0.0 a 10.255.255.255 per le reti di classe A, quelli da 172.16.0.0 a 172.31.255.255 per le reti di classe B e quelli da 192.168.0.0 a 192.168.255.255 per le reti di classe C. Infatti, gli indirizzi IP di una LAN sono visibili solo dall'interno della rete, ed è il router a mettere in contatto questa rete con l'esterno.

14.1.6 Indirizzi pubblici e privati

Abbiamo appena constatato che gli indirizzi IP possono essere pubblici o privati. I primi vengono assegnati all'host dall'Internet Service Provider in modo automatico e univoco, nel momento in cui si stabilisce la connessione. Questo accade in genere a tutti i privati che si connettono a Internet. Di conseguenza, non è possibile che due computer connessi a Internet abbiano lo stesso indirizzo IP.

Se il computer è collegato anche a una LAN (ovvero funge da gateway), disporrà in aggiunta di un indirizzo IP privato. All'interno di una rete non possono coesistere IP privati uguali, perché ogni host (anzi ogni scheda di rete, poiché in un computer possono esserne presenti più d'una) ha un suo specifico indirizzo IP. Possono essere invece presenti tanti IP privati uguali assegnati ai computer qualora questi ultimi appartengano a (sotto)reti diverse.

Entrambe le tipologie di indirizzo IP, pubblico e privato, possono essere a loro volta dinamiche o statiche. Gli indirizzi IP pubblici sono in genere dinamici, cioè cambiano a ogni connessione. In alcuni casi, gli ISP assegnano IP statici, tali cioè da non variare a ogni connessione, ma da rimanere immutati: questi IP vengono in genere offerti come servizio a pagamento e sono richiesti nel caso in cui si debba offrire un servizio Internet, ad esempio per una macchina su cui è installato un server web.

14.1.7 Comunicazione fra reti e sottoreti

Una rete locale è chiusa e non comunica con l'esterno. Per metterla in contatto con altre reti serve un dispositivo per il corretto instradamento dei pacchetti: esso prende il nome di «gateway».

Una rete locale (fisica) può essere distinta in più sottoreti logiche attribuendo alle sottoreti i bit dell'indirizzo IP relativi agli host. Ad esempio, la rete di indirizzo 192.168.x.x può essere suddivisa in due sottoreti 192.168.1.x e 192.168.2.x. La prima avrà 192.168.1.0 come indirizzo di rete e 192.168.1.255 come indirizzo di broadcast (quest'ultimo è un indirizzo speciale attraverso il quale si raggiunge ogni host della rete in questione), la seconda avrà 192.168.2.0 come indirizzo di rete e 192.168.2.255 come indirizzo di broadcast.

Per distinguere, negli indirizzi della sottorete, quali bit identificano la rete e quali l'host, si adotta per l'appunto la maschera di rete (o subnet mask). Ogni rete o sottorete è infatti caratterizzata da una stringa di 32 bit, in cui i primi, impostati a 1, identificano la rete e i secondi, impostati a 0, identificano l'host; grazie a questo sistema i router, che sono delle macchine sparse per Internet e si occupano dell'instradamento a livello IP, riescono a calcolare il percorso dei pacchetti.

14.1.8 La risoluzione dei nomi

Come abbiamo visto, la trasmissione di dati si vale degli indirizzi IP per identificare gli host, ma la loro memorizzazione risulta eccessivamente difficile. Se tutti ricordano perfettamente e senza sforzo indirizzi come <http://www.slackware.com> o <http://www.google.it>, pochi ricorderebbero i rispettivi IP, e cioè 64.57.102.34 e 66.249.93.104. Ecco perché agli indirizzi numerici sono attribuiti uno o più nomi (e viceversa, a uno stesso nome possono essere attribuiti più indirizzi IP). Deve pertanto esistere un sistema che traduce gli indirizzi in forma di nome in indirizzi in forma di numero, che opera ad esempio quando digitiamo un determinato indirizzo nella barra del browser: si tratta del DNS, acronimo di Domain Name Server. Il sistema dei nomi è organizzato ad albero e contempla domini e sottodomini; è di conseguenza possibile avere anche nomi di host identici, ma collocati in domini o sottodomini diversi, esattamente come in un file system si possono avere file dal nome uguale, a condizione che si trovino in directory o subdirectory diverse.

La risoluzione dei nomi si effettua mediante il file `/etc/hosts`, che contiene una mappa dei nomi e dei corrispondenti indirizzi IP numerici, oppure mediante un server DNS, in Slackware il demone `/usr/sbin/named`, gestito da `/etc/rc.d/rc.bind` (uno degli script che in effetti abbiamo scelto di attivare nel corso delle impostazioni del sistema al momento dell'installazione), il quale a sua volta legge le impostazioni di `/etc/named.conf`.

14.1.9 Risoluzione statica e dinamica

La risoluzione statica si basa sull'assegnazione univoca di un indirizzo IP a un nome host in `/etc/hosts` senza far ricorso a un server DNS. Questa soluzione presenta ovviamente vantaggi e svantaggi. Tra i primi annoveriamo innanzitutto la possibilità di associare un indirizzo IP a un nome host anche all'interno di una LAN non raggiungibile da Internet, laddove i server DNS riconoscono solo indirizzi IP pubblici, e secondariamente la velocizzazione del processo di risoluzione. Infatti in questo caso l'assegnazione viene effettuata semplicemente leggendo il file `/etc/hosts` presente nel computer, mentre l'operazione via

DNS implica che si effettui una chiamata o a Internet o a un server via Intranet, alla quale chiamata bisogna poi attendere la risposta contenente l'indirizzo IP risolto. Fra gli svantaggi annoveriamo innanzitutto il fatto che il file `/etc/hosts` è statico, per cui, se cambia l'indirizzo IP di un host, esso dev'essere ogni volta aggiornato, e quindi la circostanza per cui tale file non consente di applicare impostazioni DNS avanzate, come i campi MX (il record «Mail eXchange» atto alla trasmissione della posta elettronica) o PTR (il record «PoinTeR» utile per il lookup inverso).

La risoluzione dinamica si effettua mediante una chiamata al DNS cui si comunica l'indirizzo (in forma di nome) a cui ci si intende connettere. Il server risponde inviando la traduzione in forma numerica, cioè l'indirizzo IP, e di conseguenza ora l'host può finalmente connettersi. Per non incorrere in un circolo vizioso, è necessario conoscere preliminarmente l'indirizzo del server DNS in forma numerica: ecco perché i provider forniscono l'IP dei server DNS che mettono a disposizione. Questi indirizzi sono contenuti nel file `/etc/resolv.conf`.

14.2 I dispositivi di rete

14.2.1 Informazioni sui dispositivi

La scheda di rete dev'essere ovviamente riconosciuta dal computer. Il kernel deve supportarla, così come dev'essere configurato per gestire il protocollo TCP/IP. Per ottenere questa (e altre informazioni sull'hardware) ci si può valere del comando `lspci`. Per filtrare il risultato e ottenere esclusivamente i dati sulle schede di rete si ricorre a `grep`. Il comando sarà pertanto: `lspci | grep Ethernet`, che potrebbe offrire un output come:

```
root@darkstar:~# lspci | grep Ethernet
03:00.0 Ethernet controller: Broadcom Corporation NetXtreme BCM5751 Gigabit
Ethernet PCI Express (rev 01)
04:05.0 Ethernet controller: Marvell Technology Group Ltd. 88E8001 Gigabit
Ethernet Controller (rev 13).
```

I tre numeri all'inizio di ogni riga identificano i dispositivi fornendo nell'ordine bus, slot e funzione. A questo punto possiamo affinare la ricerca di informazioni su una singola scheda, col comando: `lspci -vvs 03:00.0` (dove l'opzione `-vv` predispone alla massima informatività e `-s` consente di specificare un determinato dispositivo). Si otterrà così il seguente output:

```
root@darkstar:~# lspci -vvs 03:00.0
03:00.0 Ethernet controller: Broadcom Corporation NetXtreme BCM5751 Gigabit
Ethernet PCI Express (rev 01)
Subsystem: Giga-byte Technology: Unknown device e000
Control: I/O- Mem+ BusMaster+ SpecCycle- MemWINV- VGASnoop- ParErr-
Stepping- SERR- FastB2B-
Status: Cap+ 66Mhz- UDF- FastB2B- ParErr- DEVSEL=fast >TAbort- <TAbort-
<MAbort- >SERR- <PERR-
Latency: 0, cache line size 08
Interrupt: pin A routed to IRQ 17
Region 0: Memory at f2000000 (64-bit, non-prefetchable) [size=64K]
Capabilities: [48] Power Management version 2
Flags: PMEClk- DSI- D1- D2- AuxCurrent=0mA PME(D0-,D1-,D2-,D3hot+,D3cold+)
```

```
Status: D0 PME-Enable- DSel=0 DScale=1 PME-
Capabilities: [50] Vital Product Data
Capabilities: [58] Message Signalled Interrupts: 64bit+ Queue=0/3 Enable-
Address: 44f761a33f2fcaec Data: cadf
Capabilities: [d0] #10 [0001]
```

Molti di questi dati sono comunque superflui per la semplice configurazione del dispositivo.

14.2.2 Configurazione di un dispositivo di rete

Ogni interfaccia di rete ha un proprio indirizzo fisico univoco (MAC address) che è «hardcored», cioè assegnato dal produttore e non modificabile; esso serve per identificare la periferica e perché essa possa comunicare fisicamente con il computer. È composto da sei cifre di caratteri esadecimale, ad esempio 00:0F:EA:7A:29:16, e il comando `ifconfig` lo restituisce alla voce `Hwaddr`.

Tuttavia, per rendere funzionante il protocollo IP, è necessario assegnare in aggiunta, all'interfaccia stessa, un indirizzo IP, che dev'essere compatibile con l'host. Se la rete ha, ad esempio, un indirizzo come 192.168.100.0/24 (siamo quindi all'interno di una rete di classe C), l'host dovrà assumere un indirizzo fra 192.168.100.1 e 192.168.100.254.

Si può preliminarmente verificare l'avvenuto riconoscimento della scheda di rete col comando: `dmesg | grep eth0` che dovrebbe restituire un output del tipo:

```
eth0: RealTek RTL8139 at 0xe000, 00:0d:88:1a:6c:d1, IRQ 17
eth0: Identified 8139 chip type 'RTL-8100B/8139D'
eth0: link up, 100Mbps, full-duplex, lpa 0x41E1
```

Un classico metodo per attribuire l'indirizzo IP a una scheda è ricorrere a `ifconfig`. Se lanciato senza opzioni, `ifconfig` mostra le schede già configurate. Se l'indirizzo non è stato attribuito, `ifconfig` mostra comunque che esiste per l'host un indirizzo virtuale assegnato dal sistema, e cioè l'indirizzo di `loopback`:

```
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:218 errors:0 dropped:0 overruns:0 frame:0
            TX packets:218 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:0
            RX bytes:45884 (44.8 KiB)  TX bytes:45884 (44.8 KiB)
```

Se lanciato con l'opzione `-a`, `ifconfig` mostra anche le schede non ancora configurate. In questo caso, l'output sarà:

```
root@darkstar:~# ifconfig -a
eth0       Link encap:Ethernet  HWaddr 00:0F:EA:7A:29:16
            UP BROADCAST MULTICAST  MTU:1500  Metric:1
```

```

RX packets:0 errors:0 dropped:0 overruns:0 frame:0
TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1000
RX bytes:0 (0.0 b) TX bytes:0 (0.0 b)
Interrupt:17

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:218 errors:0 dropped:0 overruns:0 frame:0
      TX packets:218 errors:0 dropped:0 overruns:0 carrier:0
      collisions:0 txqueuelen:0
      RX bytes:45884 (44.8 KiB) TX bytes:45884 (44.8 KiB)

```

Per attribuire l'indirizzo IP, va dato `ifconfig` seguito dall'indirizzo stesso ed eventualmente dall'opzione `up` per attivare immediatamente la scheda, ad esempio `ifconfig eth0 192.168.0.2 up`. Inoltre, se si usa la notazione CIDR e la maschera e il broadcast address non corrispondono a quella stabilita dal sistema `classfull`, bisogna specificarli utilizzando rispettivamente l'opzione `netmask` e `broadcast`. Dopo questa operazione, si vedrà come `ifconfig` fornisce anche l'indirizzo IP appena attribuito:

```

eth0   Link encap:Ethernet HWaddr 00:0F:EA:7A:29:14
      inet addr:192.168.0.2 Bcast:192.168.0.255 Mask:255.255.255.0
      BROADCAST MULTICAST MTU:1500 Metric:1
      RX packets:0 errors:0 dropped:0 overruns:0 frame:0
      TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
      collisions:0 txqueuelen:1000
      RX bytes:0 (0.0 b) TX bytes:0 (0.0 b)
      Interrupt:18 Memory:f1000000-0

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:218 errors:0 dropped:0 overruns:0 frame:0
      TX packets:218 errors:0 dropped:0 overruns:0 carrier:0
      collisions:0 txqueuelen:0
      RX bytes:45884 (44.8 KiB) TX bytes:45884 (44.8 KiB)

```

Questo comando consente inoltre una precisa ispezione del funzionamento delle interfacce di rete. Se infatti leggessimo fra l'altro, a proposito di `eth0`:

```

RX packets:61925 errors:0 dropped:0 overruns:0 frame:0
TX packets:72885 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1000
RX bytes:1761146 (1.5 Mib) TX bytes: 1671146 (1.5 Mib)

```

sapremmo quanti pacchetti sono stati inviati e quanti ricevuti, e quanti MB sono passati in trasmissione e in ricezione.

Questa impostazione da console non è tuttavia persistente: va infatti persa allo spegnimento del sistema. Per far sì che essa diventi permanente si ricorre a un file di configurazione: `/etc/rc.d/rc.inet1.conf`.

Durante il processo di inizializzazione di Slackware, lo script `/etc/rc.d/rc.M` richiama fra l'altro lo script `/etc/rc.d/rc.inet1` e quindi, dopo `hotplug` che ha il compito di rilevare l'hardware e dunque anche le varie interfacce di rete, `/etc/rc.d/rc.inet2`. Lo script `/etc/rc.d/rc.inet2` si occupa successivamente di avviare tutti i processi di rete: il servizio NFS, Samba, eventuali firewall, il supporto al protocollo ipv6, il server `inetd` per la gestione delle interfacce di rete, il demone `ssh`, il demone per `BIND`, il servizio `NIS` e, a sua volta, invoca il demone `inetd`. Più in dettaglio, il file `/etc/rc.d/rc.inet1.conf` viene letto durante l'inizializzazione del sistema a seguito dei comandi impartiti dallo script `/etc/rc.d/rc.inet1`.

Quest'ultimo imposta le interfacce in base alle informazioni lette appunto in `/etc/rc.d/rc.inet1.conf` ed eventualmente attiva l'interfaccia di `loopback`; quindi attiva le funzioni delle varie interfacce, verificando se deve ricorrere al kernel o ai moduli; infine attiva le funzioni di gateway, se necessario. Nel file `/etc/rc.d/rc.inet1.conf` troviamo la configurazione delle varie interfacce di rete: l'indirizzo IP se statico, il ricorso a DHCP se dinamico (in questo caso, le impostazioni DHCP sovrascrivono qualsiasi altra impostazione). Le varie interfacce che corrispondono alle schede di rete, se non precisato altrimenti, vengono chiamate `eth`, e dunque `eth0` la prima, `eth1` la seconda e così via, fino a un massimo di sei; tale numero può essere però aumentato a piacere, se il computer – caso peraltro assai strano – ne presentasse più di sei. Se c'è l'esigenza di modificare tale nome, si ricorre al campo `IFNAME`. Se si intende ricorrere all'attribuzione dinamica degli indirizzi, la voce `USE_DHCP[0]=""` dev'essere impostata a `yes`:

```
# Config information for eth0:
IPADDR[0]="192.168.0.2"
NETMASK[0]="255.255.255.0"
USE_DHCP[0]=""
DHCP_HOSTNAME[0]=""

# Config information for eth1:
IPADDR[1]=""
NETMASK[1]=""
USE_DHCP[1]=""
DHCP_HOSTNAME[1]=""

# Config information for eth2:
IPADDR[2]=""
NETMASK[2]=""
USE_DHCP[2]=""
DHCP_HOSTNAME[2]=""

# Config information for eth3:
IPADDR[3]=""
NETMASK[3]=""
USE_DHCP[3]=""
DHCP_HOSTNAME[3]=""

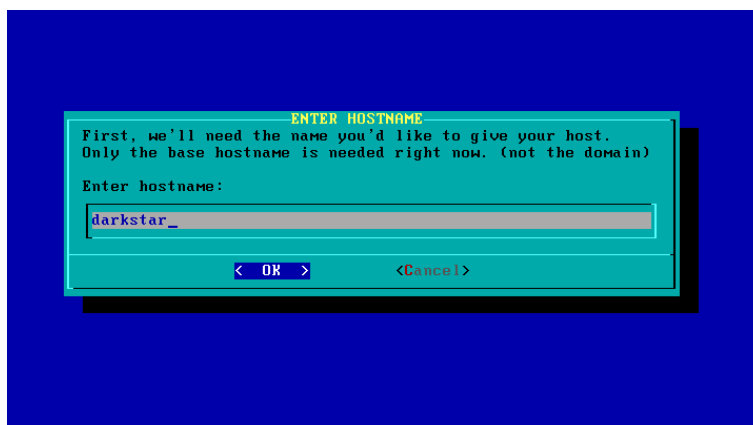
# Default gateway IP address:
GATEWAY="192.168.0.1"
```

Tutte queste operazioni di configurazioni possono essere svolte in modo facile e celere, senza mettere mano direttamente ai file di configurazione, grazie all'utilità `netconfig`.

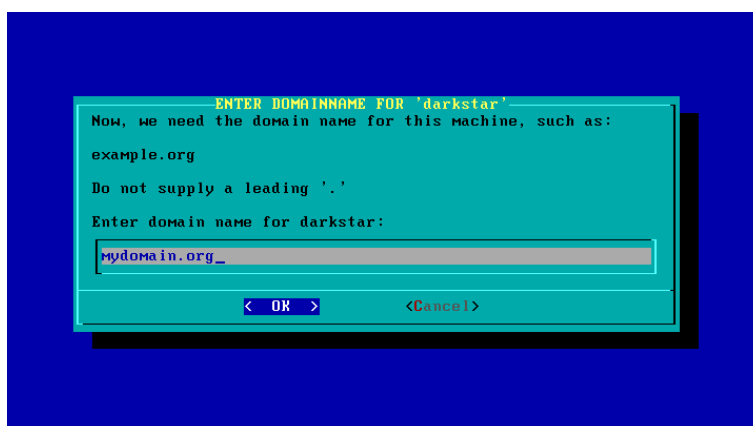
14.2.3 L'utilità netconfig

Slackware presenta una comodissima utilità pseudo-grafica per settare le interfacce di rete: `netconfig`. Essa è in genere sufficiente per tutte le esigenze normali, sia che il computer sia dotato di una scheda di rete e di un router sia di un modem seriale.

Dobbiamo innanzitutto inserire nel menu pseudo-grafico uno HOSTNAME (quello di default in Slackware è `darkstar`, il nome di uno dei computer di P. J. Volkerding):

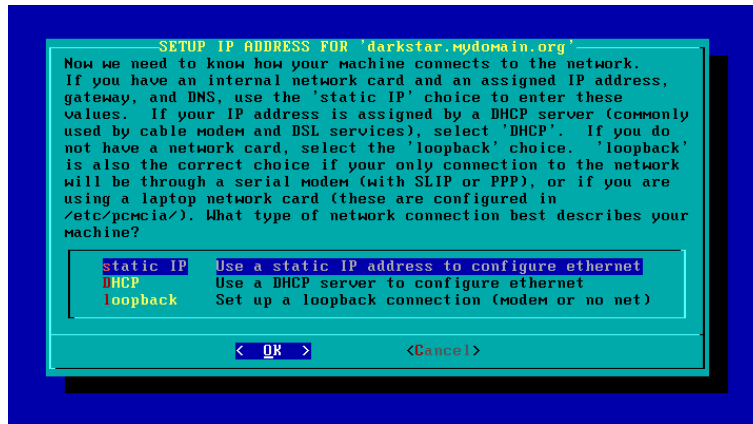


e un DOMAIN NAME a piacere (per ovvi motivi, è vivamente sconsigliato scegliere il nome di un dominio esistente; a parte la remota possibilità di conseguenze legali, potrebbe essere difficile risolvere l'indirizzo di quel dominio). Di conseguenza è conveniente optare per un nome come «`mydomain.org`»; ciò appare tuttavia vagamente insensato se in realtà non si possiede un dominio; una scelta opportuna potrebbe essere allora qualcosa come «`localdomain`». Hostname e domainname identificheranno il computer:

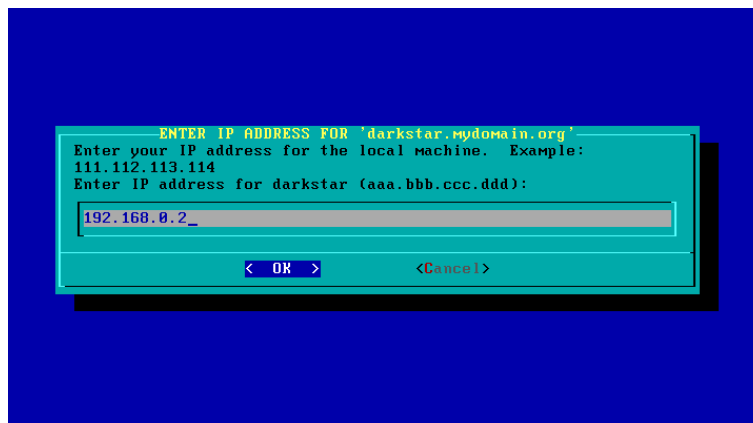


I nomi attribuiti possono essere cambiati con facilità. Il nome dell'host si può riassegnare col comando: `hostname nuovo_nome`. L'indirizzo completo (host e dominio) sono registrati in `/etc/HOSTNAME`.

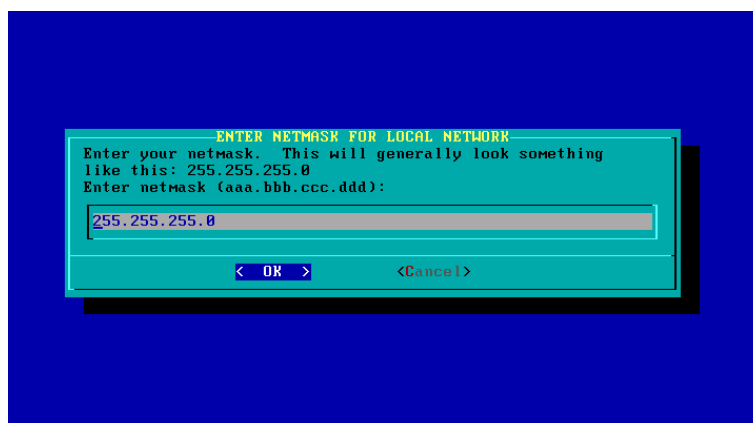
Va quindi inserito l'indirizzo IP. Il programma di installazione offre tre alternative: **static IP** (se possediamo una scheda di rete e decidiamo di attribuirvi un indirizzo IP fisso), **DHCP** (se l'IP ci viene assegnato in automatico da un server), **loopback**, se il computer non ha una scheda di rete:



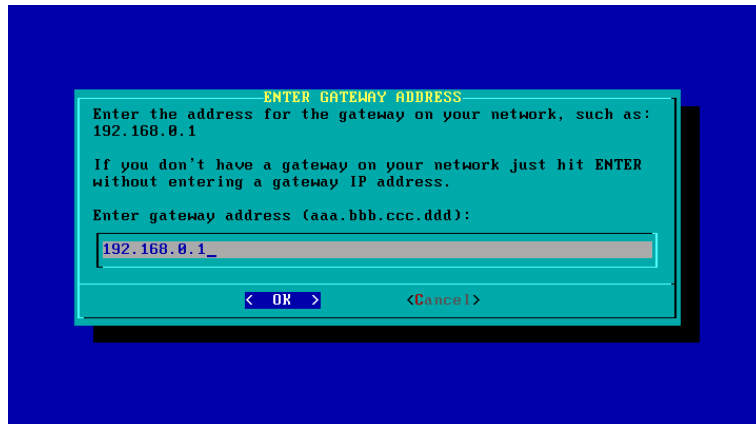
Va ora fornito l'indirizzo della nostra scheda di rete, qualcosa come 192.168.0.2:



È ora la volta della netmask, che di default per le reti di classe C è 255.255.255.0:



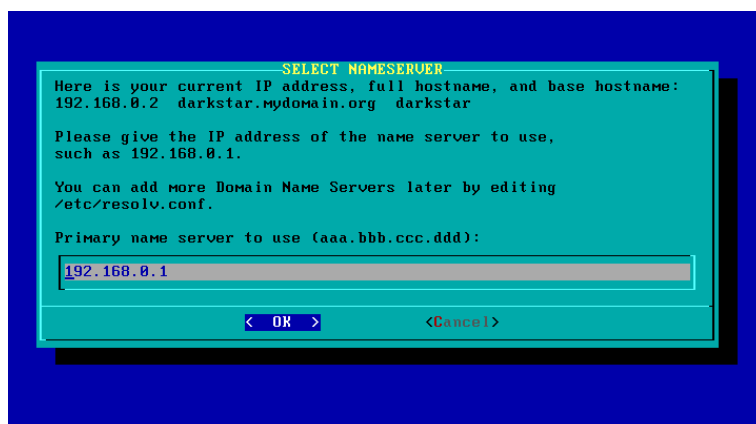
Quindi il gateway (l'indirizzo del router, come 192.168.0.1):



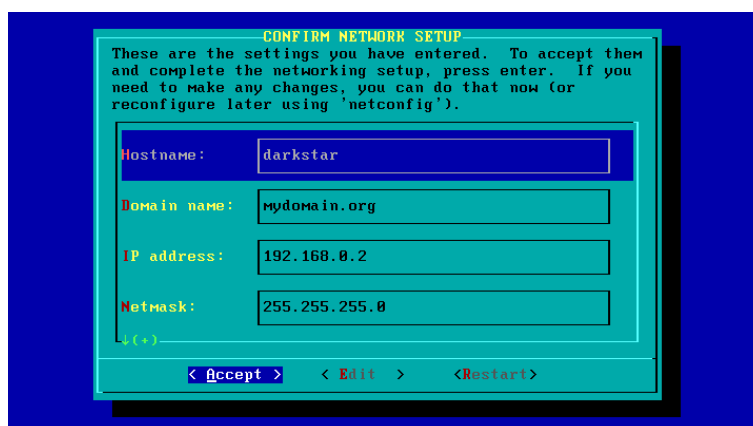
Decidiamo di abilitare un name server:



e attribuiamo il nome a questo name server, che sarà ancora una volta l'indirizzo IP del router, e cioè 192.168.0.1:



Infine si possono controllare i dati immessi e accettarli:



Essi vengono scritti nel file nel file `/etc/rc.d/rc.inet1.conf`. Un limite di `netconfig` è che tale script, per la sua semplicità, configura una e una sola interfaccia di rete. Se il computer fosse dotato di due schede di rete, la seconda andrebbe perciò configurata mettendo mano ai file di configurazione con un editor.

14.2.4 Configurare il routing

Se abbiamo un computer standing alone che si connette direttamente a Internet attraverso una scheda di rete e un modem ADSL, è sufficiente indicare al sistema tale indirizzo di rete. La situazione si fa più complicata in presenza di una rete costituita da più computer, che comunicano con l'esterno mediante un router. In questo caso, infatti, bisogna differenziare le connessioni interne con quella esterna. I computer devono infatti inviare i pacchetti destinati all'esterno a uno «scambiatore», per l'appunto il router, che li reindirizza. In genere, questo compito è affiato alla direttiva `gateway`, presente in `/etc/rc.d/rc.inet1.conf`, che indica per l'appunto l'indirizzo IP di questo medio. Il cammino che i pacchetti devono seguire è indicato nella cosiddetta tabella di instradamento (o routing). Un computer potrebbe comunicare all'interno della LAN mediante `eth0`, ma essere costretto ricorrere all'interfaccia offertagli dal server (`eth1`) per comunicare con l'esterno, cioè mediante un gateway. Per gestire la tabella di instradamento, si ricorre al comando `route`.

Il comando `route` privo di opzioni (o con l'opzione `-F` che è attiva di default) mostra la tabella di routing cercando di interpretare gli indirizzi IP, mentre con l'opzione `-n` mostra la tabella di routing corrente in forma numerica:

```
root@darkstar:~# route -n
Kernel IP routing table
Destination  Gateway      Genmask      Flags  Metric  Ref  Use  Iface
192.168.0.0  0.0.0.0     255.255.255.0  U      0        0    0    eth0
192.168.0.0  0.0.0.0     255.255.255.0  U      0        0    0    eth1
127.0.0.0    0.0.0.0     255.0.0.0     U      0        0    0     lo
0.0.0.0      192.168.0.1 0.0.0.0       UG     1        0    0    eth0
```

Le colonne indicano rispettivamente:

Destination = la rete o l'host di destinazione verso cui vengono instradati i pacchetti.

Gateway = l'indirizzo del gateway (quello che sopra abbiamo definito lo «scambiatore»), se presente; se assente, viene visualizzato un asterisco.

Genmask: la netmask della rete di destinazione. La route predefinita è 0.0.0.0.

Flags = U indica che la route è attiva («up»); H indica un host come target; G indica un gateway come target; R indica in instradamento reintegrato perché gestito in maniera dinamica; D indica in instradamento configurato da un demone (o rediretto); M indica un instradamento modificato da un demone (o da una redirezione); I indica un instradamento impedito.

Metric = indica la distanza dalla destinazione, misurata in hop. Gli hop sono gli host da attraversare per giungere alla destinazione.

Ref = numero di riferimento alla regola di instradamento. Viene usato dal kernel.

Use = numero di volte che la regola è stata presa in considerazione.

I(nter)face = nome dell'interfaccia usata per l'instradamento. Qui abbiamo due schede di rete, eth0 ed eth1.

Nella prima riga del nostro esempio, essendo la rete 192.168.0.0 direttamente connessa all'host in questione, le informazioni destinate a qualche altro nodo di quella rete possono essere inviate direttamente al nodo stesso, come si evince dal fatto che il gateway è 0.0.0.0.

Nell'ultima riga la rete di destinazione è caratterizzata dall'indirizzo 0.0.0.0, il che significa che la regola di instradamento vale per qualsiasi rete di destinazione che non sia stata esplicitamente definita altrimenti in precedenza. Ovviamente, in questo caso bisogna segnalare un gateway o un'interfaccia che si fa carico di inviare tutti i pacchetti non destinati per le rotte direttamente connesse. Ecco perché il secondo campo fornisce l'indirizzo del router (qui 192.168.0.1), che dev'essere raggiungibile (nel nostro esempio lo è perché si tratta di un nodo della rete 192.168.0.0/24, che è direttamente connessa). Non a caso, il flag non è soltanto U (che sta a indicare che il percorso è attivo), ma anche G (che identifica un gateway). Anche la maschera di rete resta indefinita in quanto ininfluente.

Per definire un instradamento nuovo si usa il comando `route` con l'opzione `add` e quindi `-net` se la destinazione (target) è una rete, oppure `-host` se la destinazione è, appunto, un host. Ad esempio, il comando: `route add -net 192.168.100.0 netmask 255.255.255.0 eth0` definisce l'instradamento per l'interfaccia `eth0` attraverso la rete 192.168.0.x. Per aggiungere l'instradamento predefinito per `eth0` via gateway il comando sarà: `route add default gw 192.168.255.0 eth0`.

Infine, per azzerare la tabella di reinstradamento, ripristinando quella gestita direttamente dal kernel, si usa il comando `route` con l'opzione `del` seguita dal numero della rete da eliminare, ad esempio: `route del 192.168.100.0`.

14.2.5 Configurare un server DHCP/DNS

Molti modem/router includono una funzionalità di server DHCP (acronimo di «Dynamic Host Configuration Protocol»). Sono così in grado di svolgere automaticamente una serie di funzioni che mettono l'inesperto al riparo da configurazioni più o meno complesse: fondamentalmente, assegnano a ciascun client i parametri per la connessione (indirizzo IP, netmask, server DNS); l'indirizzo IP è a volte addirittura stabile perché impostato in base al MAC address della scheda di rete. È perciò sufficiente connettere l'hardware per trovare già un sistema funzionante.

Se invece il router non è dotato di questa funzionalità, soccorrono alcuni programmi che sono in grado di erogarla. In Slackware i pacchetti che sovrintendono al server DHCP sono **dhcp** (contenente il server e le utilità client) e **dhcpcd** (che contiene il demone client), che presentano alcune differenze di struttura e di impostazione rispetto ad altre distribuzioni GNU/Linux.

Componenti del protocollo

Il protocollo DHCP si sostanzia di tre fondamentali componenti:

il **client DHCP**, che identifica da un lato il computer che ha bisogno di ottenere un indirizzo IP valido per la sottorete a cui è collegato, e dall'altro anche il programma che si occupa di richiedere l'indirizzo IP e di configurarlo

il **server DHCP**, che identifica da un lato il computer che assegna gli indirizzi IP, e dall'altro anche il processo che svolge questa funzione

il **DHCP relay**, ossia il computer (o il router) che si occupa di inoltrare le richieste DHCP a un server, qualora questo non sia sulla stessa sottorete. Questo componente è necessario solo se un server DHCP deve servire innumerevoli sottoreti. Deve esistere almeno un DHCP relay per ciascuna sottorete servita e ogni relay dev'essere esplicitamente configurato per inoltrare le richieste a uno o più server.

Funzionamento del protocollo

DHCP utilizza il protocollo UDP; le porte registrate sono la 67 per il server e la 68 per il client.

Quando un computer vuole ottenere un indirizzo tramite DHCP, attiva il processo DHCP client. In questo momento, esso non possiede un indirizzo IP valido, e di conseguenza non può usare tutte le funzionalità della rete.

Quindi esso invia un pacchetto chiamato DHCPDISCOVER in broadcast nella sottorete, con indirizzo IP sorgente impostato convenzionalmente a 0.0.0.0 (che significa in pratica un indirizzo qualsiasi), e con destinazione 255.255.255.255. Tutti i server DHCP attivi nella sottorete ricevono questo pacchetto e possono rispondere (o meno) con un pacchetto di DHCPOFFER, in cui assegnano un indirizzo IP al client. Il pacchetto è indirizzato direttamente all'indirizzo di livello `livello_dataLink` del client (che non ha ancora un indirizzo IP), per cui può essere inviato solo da un server che si trovi sulla stessa sottorete.

Se nella sottorete ci sono più DHCP relay, questi inoltrano il pacchetto al loro server di riferimento, che può rispondere allo stesso modo attraverso il relay. Il relay agent informa il server della sottorete da cui ha ricevuto il pacchetto di DHCPDISCOVER, permettendo al server di offrire un indirizzo per una sottorete valida.

Il client attende un certo tempo per ricevere una o più offerte, dopo di che ne seleziona una e invia un pacchetto di DHCPREQUEST al server che ha scelto. Questo gli conferma l'assegnazione dell'indirizzo con un pacchetto di DHCPACK.

Scadenza e rinnovo degli indirizzi

A questo punto, il client è autorizzato a usare l'indirizzo ricevuto per un tempo limitato, detto «tempo di lease». Esiste un tempo di lease predefinito (ventiquattr'ore) che può essere modificato a piacere dall'amministratore del server DHCP. Prima della scadenza, il client dovrà tentare di rinnovarlo inviando un nuovo pacchetto DHCPREQUEST al server che, se intende prolungare l'assegnazione dell'indirizzo, gli risponderà con un nuovo DHCPACK. Se il client non riesce a rinnovare l'indirizzo, tornerà allo stato iniziale cercando di farsene attribuire uno nuovo.

Identificazione e autenticazione dei client

Il client si identifica nei confronti del server mediante un campo `client-id` dei pacchetti DHCP, che di solito ha come valore il MAC address della scheda di rete per cui si richiede l'indirizzo (ma che potrebbe anche essere configurato manualmente). Ciò implica una falla nella sicurezza, perché tale dato, inviato in broadcast nella sottorete, può essere facilmente sniffato da qualunque altro calcolatore connesso alla sottorete. Vi sono dei sistemi più avanzati di sicurezza per controllare l'accesso alla rete, che però richiedono un supporto da parte degli switch a cui sono collegati gli host (come il progetto IEEE802.1x, volto a definire gli standard per l'architettura delle LAN). Il server può utilizzare il campo `client-id` per decidere se rispondere al client, quale indirizzo assegnargli e quali altri parametri eventualmente passargli.

Identificazione del server e sicurezza

Il server si identifica nei confronti del client con il proprio indirizzo IP. Un client può di conseguenza essere impostato in modo da accettare indirizzi IP solo da un server già noto.

Ogni computer collegato alla sottorete potrebbe fungere da server DHCP per gli altri di quella sottorete, o da relay verso un server DHCP arbitrario. Un server mal configurato, oppure uno artatamente configurato in modo malizioso, è dunque in grado di offrire indirizzi IP creando malfunzionamenti alla rete e/o problemi di sicurezza.

Un calcolatore che abbia ricevuto l'indirizzo IP da un server DHCP mal configurato non è in grado di accedere alla rete. Ma un calcolatore che abbia ricevuto l'indirizzo IP da un server DHCP malizioso può offrire indirizzi che sa essere inutilizzati, oppure collocati in una sottorete IP diversa da quella ufficiale, senza produrre conflitti con il server autentico, indicando se stesso come default gateway. Quindi sarà posto nelle condizioni di ridirigere le connessioni effettuate dai client verso il gateway ufficiale utilizzando l'IP masquerading, tanto da intercettare e sniffare tutto il traffico generato dai client, che difficilmente potrebbero accorgersi della differenza.

Opportunità di adottare un server DHCP

Abbiamo premesso come l'adozione di un sistema DHCP per conferire dinamicamente gli indirizzi IP si mostra molto funzionale nelle reti domestiche prive di problemi di sicurezza, se non altro perché risparmia all'utente tutti i problemi di configurazione della rete.

Tuttavia, indipendentemente dal fatto che si utilizzi o meno il protocollo DHCP, alcuni indirizzi IP devono sempre rimanere statici. Si tratta degli host che eseguono i protocolli DHCP, WINS e DNS (così come è quanto meno consigliabile un indirizzo IP statico per i Domain Controller). Anche i server WEB e SMTP devono avere un indirizzo statico: i primi perché tale indirizzo sarà pubblico e registrato nei DNS dei vari provider (e, in aggiunta, qualora il server WEB sia dietro un firewall, l'indirizzo deve comunque essere fisso perché va impostato nelle tabelle di PAT / NAT del firewall), i secondi perché i loro indirizzi devono andare nei record MX dei DNS (per la gestione della posta elettronica). Analoga situazione per un server con funzioni di database (i server SQL server e/o di MySql), perché tali servizi sono di solito utilizzati da programmi che accedono ai dati via IP.

La configurazione del server

Innanzitutto il server deve sapere su quali interfacce di rete rimanere in ascolto. Esse vengono indicate, al solito, in `/etc/rc.d/rc.inet1.conf`.

In secondo luogo, il server deve sapere quale configurazione si intende fornire al client; essa viene specificata in `/etc/dhcpd.conf`. La sintassi di questo file si compone di parametri e dichiarazioni. I parametri dicono come fare qualcosa (ad esempio per quanto tempo deve durare un'assegnazione), se fare qualcosa (ad esempio, se fornire o meno indirizzi a client sconosciuti), quali parametri fornire ai client (ad esempio, quale gateway usare). Le dichiarazioni si usano per descrivere la topologia della rete, i client, per stabilire gli indirizzi che andranno assegnati ai client, per applicare gruppi di parametri a gruppi di dichiarazioni.

Per ogni sottorete, deve comparire in `/etc/dhcpd.conf` una dichiarazione di sottorete (subnet) che suggerisce a dhcpd come riconoscere che un determinato indirizzo appartiene a quella sottorete. Ogni sottorete deve avere i suoi parametri specificati in forma distinta, perché ciascuna ha (o può avere) il suo router, indicato con un'apposita stringa:

```
option routers 204.254.239.1;
```

Se ai client di una sottorete gli indirizzi IP devono essere assegnati in forma dinamica, all'interno della dichiarazione di sottorete non deve mancare la dichiarazione dell'intervallo di indirizzi. Per i client

a cui va assegnato un indirizzo IP in forma statica, dev'esserci una dichiarazione distinta per ogni host. Quando dhcpd trova una dichiarazione per un client, prima verifica se gli è stato attribuito un indirizzo fisso e, se non lo trova, passa alla dichiarazione che serve ad attribuire dinamicamente gli indirizzi.

I parametri globali vengono passati all'inizio del file; potrebbe trattarsi del nome di dominio dell'organizzazione, dell'indirizzo dei DNS ecc., ad esempio:

```
option domain-name "isc.org";
option domain-name-servers ns1.isc.org, ns2.isc.org;
```

Si possono fornire sempre e solo indirizzi dinamici a tutti gli host, oppure, per immaginare uno degli infiniti scenari possibili (ma prossimo a un utilizzo all'interno di una piccola rete privata), si potrebbe ipotizzare di scegliere l'attribuzione fissa degli indirizzi per le periferiche di rete stabili, e dinamica per delle periferiche che vengono rimosse spesso, oppure che si connettono solo occasionalmente al server, come dei computer portatili (si tratta del cosiddetto «indirizzamento misto»).

Nel primo caso il file /etc/dhcpd.conf avrà una struttura estremamente semplice, ad esempio:

```
subnet 192.168.10.0 netmask 255.255.255.0 {
    range 192.168.10.10 192.252.10.250;
}
```

Nel secondo caso, le opzioni pertinenti potrebbero essere:

```
option domain-name 'lan.domestica'
option domain.name-servers 212.216.112.222, 212.216.172.162, 212.245.255.2
default-lease-time 600
max-lease-time 7200
log-facility local7
authoritative
subnet 192.168.10.0 netmask 255.255.255.0 {
range 192.168.10.2 192.168.10.100;
range 192.168.10.150 192.168.10.254;
option routers 192.168.10.1;
}
host samiel {
hardware ethernet 00:0F:EA:7A:29:16
fixed-address 192.168.0.151
}
```

In questo modo, ai computer che fanno richiesta di configurazione di rete verrà attribuito un dominio valido e forniti al tempo stesso gli IP di alcuni DNS (prime due righe). L'assegnazione fin qui effettuata potrebbe però portare a uno spreco di risorse se alcuni dei computer connessi fossero dei portatili, soggetti perciò spesso a essere disconnessi dal sistema, in quanto gli IP attribuiti resterebbero assegnati, ancorché inutilizzati. A ciò si ovvia istituendo un tempo predefinito (espresso in secondi) dopo il quale l'assegnazione scade e dev'essere rinnovata, e un tempo massimo dopo il quale essa non può più essere

rinnovata (terza e quarta riga). Di default, le assegnazioni valgono per un giorno, dopo di che quell'indirizzo non può più essere attribuito allo stesso host; viene conservata notizia delle attribuzioni effettuate in `/var/state/dhcp/dhcpd.leases`, file che `dhcpd` legge dopo `/etc/dhcpd.conf`.

È anche possibile formare dei blocchi di configurazione differenziati per effettuare nei confronti di utenti diversi assegnazioni con valori temporali diversi, ricorrendo al logger di sistema (quinta riga). Per indirizzare i file di log del server verso uno specifico file bisogna aggiungere nel file `/etc/syslog.conf` una riga quale:

```
local7.debug /var/log/dhcpd.log
```

Qui però prescindiamo da questa più complessa configurazione. Ampie spiegazioni e alcuni esempi sono reperibili alla pagina `man` di `dhcpd.conf`.

La stringa `authoritative` (sesta riga) serve se e solo se nella stessa rete sono presenti più server DHCP, perché, nel caso in cui un client richieda un indirizzo IP non adatto per la sua sottorete, il DHCP server possa avere l'autorità per negarli la richiesta.

Il blocco di configurazione successivo (dalla settima all'undicesima riga) gestisce l'assegnazione degli IP ai computer connessi. La prima stringa definisce l'indirizzo della rete relativamente alla quale sono forniti gli indirizzi IP e la relativa `submask`. La scelta degli indirizzi è del tutto opzionale; l'unico elemento imprescindibile è che sussista coerenza fra l'indirizzo della rete, la `netmask` e gli indirizzi dei singoli host. Le parentesi graffe definiscono le istruzioni atte ad attribuire in dettaglio gli IP ai vari host: essi saranno compresi negli intervalli (`1textitrange`) indicati, mentre gli IP compresi nell'intervallo da `.101` a `.149`, che non sono indicati, non verranno assegnati e resteranno disponibili per assegnazioni statiche. Anche in questo caso le scelte sono largamente opzionali, fatto salvo che non dovranno essere assegnati gli indirizzi `.0` (riservato all'indirizzo della rete), `.255` (riservato all'IP di broadcast) e `.1` (qui associato al router).

Stabilito in tal modo come attribuire gli indirizzi dinamici, vediamo adesso come attribuire quegli statici. Il meccanismo si basa sul riconoscimento di un dato inalterabile come l'indirizzo MAC della scheda di rete e quindi sull'attribuzione a questo indirizzo di un indirizzo IP. Quando al server DHCP arriva una richiesta di attribuzione da parte di un dispositivo con tale indirizzo MAC, esso risponderà attribuendogli l'IP predeterminato alla voce `fixed-address`.

Questa configurazione di base e molto semplice può essere raffinata definendo un comportamento specifico per più blocchi, ai quali si possono attribuire in aggiunta delle regole in contrasto con quelle definite nei criteri generali.

14.2.6 Configurare una rete con KDE

KDE offre una comoda interfaccia grafica per effettuare le operazioni di configurazione della rete (sotto Gnome ne esiste l'analogo, `network-admin`). Essa da un lato rende assai lineare e semplice il processo, dall'altro però nasconde in effetti i vari file di configurazione a cui vengono applicate le modifiche. Questo strumento è reperibile in **Centro di controllo – Internet e rete – Impostazioni di rete**. Le voci, tutte molto intuitive, sono parte di quelle che abbiamo fin qui analizzato. Utile anche la voce **Centro di controllo – Internet e rete – Proxy**, da cui si impostano i proxy di sistema, atti a consentire la navigazione in ambienti protetti da firewall.

14.3 File di configurazione della rete

Slackware presenta diversi file di configurazione della rete tipici dei sistemi Unix, ma, data la struttura BSD dei suoi script di inizializzazione, non è priva di alcune peculiarità. Vediamo, tra questi, i

principali, quelli che (essendo semplici file di testo e non binari) possono essere modificati a seconda delle esigenze.

Molte impostazioni, anzi in genere tutte quelle che servono per navigare con un modem o con un router, vengono applicate con facilità con gli strumenti di configurazione che vedremo oltre, ma è comunque importante conoscere il ruolo, il contenuto e la sintassi di questi file, perché a volte è necessario mettervi mano direttamente.

14.3.1 Il file `/etc/hosts`

Il file `/etc/hosts` contiene la traduzione degli indirizzi IP in nomi. È atto a gestire la risoluzione dei nomi in reti piccole, prive di un server DNS. La sintassi che presenta è: [indirizzo IP] [nome] [alias]. Nella sua forma più semplice, potrebbe essere:

```
# For loopbacking.
127.0.0.1 localhost
192.168.0.2 darkstar.localdomain darkstar

# End of hosts.
```

dove vengono indicati soltanto l'indirizzo di loopback e l'IP della scheda di rete del sistema, col suo nome e il suo alias. Un esempio appena più complesso potrebbe essere:

```
# For loopbacking.
127.0.0.1 localhost
192.168.100.102 server01.mynetwork mynetwork
192.168.0.1 pc01.mynetwork pc01
192.168.1.2 pc02.mynetwork pc02
192.168.100.254 gateway.mynetwork gateway

# End of hosts.
```

Qui abbiamo a che fare con una LAN composta da un server e due client. Sarà così possibile riferirsi ai vari computer, nei file di configurazione, mediante il nome e, ancor più brevemente, con l'alias, anziché mediante l'indirizzo IP.

14.3.2 Il file `/etc/resolv.conf`

Il file `/etc/resolv.conf` serve per tradurre in indirizzi IP i nomi non presenti in `/etc/hosts`. La sua sintassi è: [parola_chiave] [valore], dove *parola_chiave* può assumere i seguenti valori: `nameserver`, il cui valore consiste nell'indirizzo IP del server da interrogare; `domain`, che fornisce il nome della rete locale; `search`, che consiste in un elenco di domini da interrogare. I vari nomi sono separati da spazi. Ecco un semplice esempio di `/etc/resolv.conf`:

```
domain localhost
nameserver 212.216.112.222
```

```
nameserver 212.245.255.2
search inwind.it
search tin.it
```

Nel nostro esempio, la voce `domain localhost` indica il dominio predefinito nel quale vengono condotte le interrogazioni; di conseguenza, se venisse indicato un nome di host privo di dominio, gli verrebbe attribuito quello qui specificato.

Le voci `nameserver 212.216.112.222` e `nameserver 212.245.255.2` precisano un determinato IP come indirizzo di un DNS. Esse vengono interrogate nell'ordine in cui sono scritte, nel caso il dominio predefinito non risponda. Ovviamente, se nessun server risponde si avrà il fallimento del processo di risoluzione e la conseguente impossibilità di navigare in Internet.

Infine, le voci `search inwind.it` e `search tin.it` servono per creare dei nomi di dominio predefinito alternativi e complementari a quello indicato alla prima voce.

14.3.3 Il file `/etc/host.conf`

Il file `/etc/host.conf` indica il metodo da usare per la risoluzione dei nomi. L'utilità di questo file appare evidente nel caso che segue. Ipotizziamo di aver indicato lo stesso dominio sia in `/etc/hosts` sia in `/etc/resolv.conf`, ma con indirizzo IP diverso. Come deve comportarsi il sistema? Ecco allora che il file `/etc/hosts` ordina le interrogazioni. Esso contiene una direttiva di forma: `[parola_chiave] [valore]`. Nella sua configurazione più semplice appare come segue:

```
order hosts, bind
multi on
```

Le parole chiave utilizzabili sono:

`order`: indica l'ordine dei metodi da utilizzare per la risoluzione. I valori possibili sono `hosts`, `bind` e `nis`. Nel nostro esempio, il file ordina di interrogare dapprima il file `/etc/hosts`, e quindi, solo se non verrà trovata alcuna associazione, di rivolgersi ai server esterni presenti in `/etc/resolv.conf`. Infatti `bind` è il servizio che gestisce il demone `named`. Infine, si rivolgerà a `NIS` (che sta per «Network Information Services»), un sistema client/server basato su `RPC` che consente a un gruppo di macchine in un dominio `NIS` di condividere un insieme di file di configurazione. Ciò permette a un amministratore di sistema di installare sistemi client `NIS` con il minimo di dati di configurazione e di aggiungere, rimuovere o modificare dati di configurazione da una singola macchina. Benché le implementazioni interne dei due sistemi siano del tutto diverse, `NIS` può essere paragonato, quanto alle funzionalità base, a un sistema di domini di `Windows NT`[®]. Qui prescindiamo completamente da questa problematica.

`multi`: indica la possibilità di avere nomi di host con diversi indirizzi IP. I valori possibili sono `on` e `off`. Il valore `on` assume tutti gli indirizzi IP associati a un medesimo host. Ciò può essere utile ad esempio quando un provider ha più server, ma non è detto che tutti siano disponibili; come contraltare, se gli indirizzi IP sono molto, può aversi un certo rallentamento nella navigazione. Il valore `off` assume solo il primo indirizzo IP disponibile.

`nospoof`: attiva o meno la funzione di esercitare un controllo incrociato sulla risoluzione dei nomi. Il nome viene tradotto in indirizzo IP e successivamente quest'ultimo viene ritradotto in nome per verificarne la coincidenza; ciò serve a evitare lo spoofing dei nomi di host.

In generale lo spoofing (che letteralmente significa «imbroglio») consiste nel simulare che un certo file appaia diverso da ciò che è (ad esempio, un virus può essere camuffato da un allegato di solo testo o da un file di immagine), oppure nell'inviare una mail che presenta un'intestazione malformata ad arte

basandosi sul protocollo MIME (che gestisce gli allegati), oppure ancora nel modificare un IP non valido per accedere a un sistema in uno che invece è autorizzato. A noi qui interessa però solo quest'ultimo caso, l'IP spoofing, di cui si riconoscono tre differenti tecniche: l'IP spoofing non cieco, l'IP spoofing cieco e di tipo DOS.

Nell'IP spoofing non cieco colui che effettua l'attacco cerca di camuffarsi come un host appartenente alla sottorete dell'attaccato: quindi i pacchetti destinati all'host risulteranno visibili anche a lui. Questa tecnica è però realizzabile solo se l'attaccante conosce determinati parametri dei pacchetti (o meglio, degli header dei pacchetti in transito: il sequencer number e l'acknowledgement number). Nello spoofing cieco invece l'attaccante deve cercare di indovinare il sequencer number dei pacchetti per continuare la connessione e l'intercettazione dei dati. Infine, gli attacchi di tipo DOS (acronimo di «Denial of Service») sono volti a bloccare un determinato host, impedendogli di svolgere le sue normali attività e di conseguenza impedendo agli utenti di poter fare accesso ai servizi che tale host era destinato a effettuare.

I valori possibili della funzione `nospoof` sono `on` e `off`.

`spoofalert`: notifica o meno al log di sistema le eventuali verifiche fallite. I valori possibili sono `on` e `off`.

`spoof`: gestisce le funzionalità di spoof. I valori possibili sono `off`, `nowarn` e `warn`. Il valore `off` autorizza gli indirizzi «spoofed» senza notificare alcunché. Il valore `nowarn` impedisce la traduzione degli indirizzi «spoofed» senza notificare alcunché. Il valore `warn` impedisce la traduzione degli indirizzi «spoofed» abilitando al contempo la notifica via `syslog`.

`reorder`: ordina i valori in modo che gli indirizzi locali e di rete locale siano letti per primi. I valori possibili sono `on` e `off`.

14.3.4 Il file `/etc/rc.d/rc.inet.1.conf`

Il file `/etc/rc.d/rc.inet.1.conf` presenta la configurazione delle interfacce di rete. Ne abbiamo già analizzato la struttura fondamentale a proposito di `ifconfig`.

14.3.5 I file `/etc/hosts.allow` e `/etc/hosts.deny`

I file `/etc/hosts.allow` e `/etc/hosts.deny` servono per consentire e vietare a determinati host di utilizzare i servizi di rete messi a disposizione dal demone `inetd`, quindi in relazione a svariati protocolli e non solo al TCP/IP. La loro sintassi è: `[demone]: [client][: comando_shell]`.

Il campo `[demone]` presenta uno o più nomi dei demoni dei processi, separati da spazi.

Il campo `[client]` presenta uno o più nomi di host (si può anche fare ricorso a degli elenchi contenuti in un altro file, precisandone il percorso assoluto completo, separati da spazi).

Oltre ai nomi, ci si può valere delle seguenti espressioni:

ALL: si applica a tutti i client in riferimento un particolare servizio oppure a tutti i servizi che usano il controllo dell'accesso. L'opzione **ALL** può essere usata anche per i demoni.

LOCAL: si applica a tutti gli host che non contengono il carattere punto (`.`).

KNOWN: si applica a ogni host il cui nome o indirizzo viene riconosciuto, oppure quando l'utente viene riconosciuto.

UNKNOWN: si applica a ogni host il cui nome o indirizzo non viene riconosciuto, oppure quando l'utente non è riconosciuto.

PARANOID: si applica quando il nome di host non corrisponde all'indirizzo di host.

Il campo `[comando_shell]` è opzionale e specifica un comando da attuare nel caso venga utilizzata la regola. Questo tipo di comando viene usato soprattutto con le regole che negano l'accesso e permettono di impostare delle trappole che solitamente registrano in un file informazioni i tentativi di accesso malizioso e/o inviano queste informazioni all'amministratore.

Nella formulazione delle regole sono ammessi anche dei caratteri jolly come il punto (.), che posto all'inizio di una stringa indica tutti i membri dell'insieme specificato: ad esempio, la stringa `.domain.com` vale sia per `system1.domain.com` sia per `system2.domain.com`.

I servizi non precisati in alcuna regola di questi due file sono consentiti a tutti. Le regole presenti in `/etc/hosts.allow` hanno la precedenza su quelle specificate in `/etc/hosts.deny`, tanto che se in `/etc/hosts.deny` è contenuta una regola che vieta l'accesso a un servizio, gli host a cui è stata data l'autorizzazione in `/etc/hosts.allow` possono comunque accedervi. Le regole specificate hanno effetto dal basso verso l'alto, per cui è rilevante l'ordine in cui sono posizionate. Le modifiche apportate ai file hanno effetto immediato e non è necessario riavviare i servizi.

Per gestire al meglio il controllo dell'accesso, `/etc/hosts.allow` e `/etc/hosts.deny` vengono usati insieme. Di norma, si vieta tutto a tutti in `/etc/hosts.deny` con `ALL` e si conferiscono le autorizzazioni specifiche in `/etc/hosts.allow`. Un esempio di accoppiata potrebbe essere:

```
/etc/hosts.deny
```

```
ALL: ALL
```

```
/etc/hosts.allow
```

```
portmap, in.telnetd: 10.0.1.24
in.ftpd: 10.0.1
```

Infine, si può ricorrere all'operatore `EXCEPT`, che consente di associare elenchi separati alla stessa regola. Quando `EXCEPT` viene usato fra due elenchi, ha effetto il primo a meno che una voce del secondo non corrisponda a un'entità del primo elenco. `EXCEPT` può essere usato sia con demoni sia con client. Potremmo allora avere un caso di `/etc/hosts.allow` come quello che segue:

```
# all domain.com hosts are allowed to connect
# to all services except cracker.domain.com
ALL: .domain.com EXCEPT cracker.domain.com

# 123.123.123.* addresses can use all services except FTP
ALL EXCEPT in.ftpd: 123.123.123.
```

14.3.6 Il file `/etc/HOSTNAME`

Il file `/etc/HOSTNAME` contiene semplicemente il nome dell'host, visualizzato col comando `hostname`. Questo può essere modificato col comando: `hostname nuovo_nome`. Analoga funzione hanno i comandi `domainname` e `ypdomainname`, che mostrano o impostano il nome di dominio NIS/YP del sistema. I comandi `dsndomainname` e `nisdomainname` si limitano a mostrare rispettivamente il nome del dominio DNS e quello del dominio NIS/YP del sistema. Questi nomi servono per l'identificazione della macchina.

14.3.7 Il file `/etc/networks`

Il file `/etc/networks` converte i nomi delle sottoreti in indirizzi di rete, con la sintassi `[nome] [indirizzo IP] [alias]`, ed è particolarmente utile al boot del sistema, allorché non è attivo nessun name server. Un semplice esempio (senza uso degli alias) è:

```
localdomain 127.0.0.0
mynetwork 192.168.100.0
```

Questo file è usato da `route` per visualizzare i nomi delle sottoreti.

14.3.8 Il file `/etc/protocols`

Il file `/etc/protocols` fornisce la mappatura dei servizi Internet secondo l'assegnazione di Iana.org.

14.3.9 Il file `/etc/services`

Il file `/etc/services` è di notevole lunghezza, perché enuncia la corrispondenza fra i servizi del sistema e la corrispondente porta di cui si valgono. La sua sintassi è `[nome_servizio] [porta/protocollo] [alias]`. Queste informazioni sono di vitale importanza quando si sta impostando un firewall. La più gran parte dei servizi ha due porte (TCP e UDP) anche se il protocollo non supporta l'UDP. Ad esempio, le righe:

```
http 80/tcp www www-http #World Wide Web HTTP
http 80/udp www www-http #World Wide Web HTTP
```

significano che il protocollo `http`, quello che serve a raggiungere Internet, è operativo sulla porta 80; bloccare questa porta ha come conseguenza l'impossibilità di navigare nel web.

Solo in rarissimi casi Slackware ha una politica diversa da quella di altre distribuzioni GNU/Linux (abbiamo visto il caso della condivisione di uno scanner), ma per lo più e generalmente ha impostazioni tali per cui tutti i protocolli per la navigazione (`http`, `ftp`, `telnet` ecc.) e per la gestione della posta elettronica (`pop3`, `smtp` ecc.) sono preimpostati in modo di consentire l'uso di questi servizi senza mettere mano al file.

14.4 Comandi per la configurazione della rete

Nei sistemi Unix esiste una serie di comandi basilari per la gestione delle reti che serve per scopi diagnostici o di configurazione. Molti di essi si rivelano utili anche soltanto per la connettività in Internet in computer standing alone. I primi due che elenchiamo sono utili a verificare il funzionamento di una generica connessione a Internet, mentre i successivi comandi sono strumenti DNS, atti cioè a verificare la corretta risoluzione dei nomi.

14.4.1 Il comando `ping`

Il comando `ping` invia una richiesta ICMP di eco alla destinazione (host) specificata mediante indirizzo o IP. Se la connessione funziona e la macchina «pingata» può rispondere, l'output mostra i pacchetti trasferiti e l'IP da cui provengono:

```
samiel@darkstar:~# ping www.google.it
PING www.l.google.com (64.233.183.147) 56(84) bytes of data.
64 bytes from 64.233.183.147: icmp_seq=1 ttl=242 time=4880 ms
64 bytes from 64.233.183.147: icmp_seq=3 ttl=242 time=4007 ms
64 bytes from 64.233.183.147: icmp_seq=4 ttl=242 time=3054 ms
64 bytes from 64.233.183.147: icmp_seq=5 ttl=242 time=3021 ms
64 bytes from 64.233.183.147: icmp_seq=6 ttl=242 time=1127 ms
64 bytes from 64.233.183.147: icmp_seq=7 ttl=242 time=1380 ms
64 bytes from 64.233.183.147: icmp_seq=8 ttl=242 time=590 ms
64 bytes from 64.233.183.147: icmp_seq=9 ttl=242 time=881 ms
64 bytes from 64.233.183.147: icmp_seq=10 ttl=242 time=244 ms
64 bytes from 64.233.183.147: icmp_seq=11 ttl=242 time=220 ms
64 bytes from 64.233.183.147: icmp_seq=12 ttl=242 time=214 ms
64 bytes from 64.233.183.147: icmp_seq=13 ttl=242 time=215 ms
64 bytes from 64.233.183.147: icmp_seq=14 ttl=242 time=215 ms
64 bytes from 64.233.183.147: icmp_seq=16 ttl=242 time=216 ms
64 bytes from 64.233.183.147: icmp_seq=17 ttl=242 time=210 ms
64 bytes from 64.233.183.147: icmp_seq=18 ttl=242 time=207 ms

[1]+  Stopped                  ping www.google.it.
```

L'operazione si stoppa col comando `Ctrl+C`, poiché per impostazione predefinita ping invia infiniti pacchetti; è però possibile specificare il numero dei pacchetti, le loro dimensioni e l'intervallo di tempo fra una richiesta e l'altra (di default, un secondo). È naturalmente possibile verificare con ping anche un host situato all'interno di una rete locale.

Il fatto che il comando fallisca dando un output come:

```
root@darkstar:~# ping www.google.it
ping: unknown host www.google.it
```

può essere dovuto a innumerevoli ragioni, non necessariamente a un'erronea configurazione del dispositivo di rete o della tabella di instradamento: esso può dipendere dalla momentanea indisponibilità dell'host target, oppure dal fatto che esso è configurato in modo da non rispondere alle richieste di ping.

14.4.2 Il comando traceroute

Il comando `traceroute` mostra i vari host che un pacchetto attraversa per raggiungere la sua destinazione. Si può così valutare il numero degli hop che ci separano da un determinato sito; terminata l'operazione, il processo si chiude da solo e si ritorna al prompt della console. Anche `traceroute`, come ping, usa pacchetti ICMP. Ecco un output di esempio:

```
samiel@darkstar:~# traceroute www.google.it
traceroute: Warning: www.google.it has multiple addresses; using 66.249.93.104
traceroute to www.l.google.com (66.249.93.104), 30 hops max, 38 byte packets
 1  x-vr1.opb.interbusiness.it (217.141.110.1) 177.637 ms 171.904 ms 166.962 ms
```

```

2 r-vr31-vl11.opb.interbusiness.it (217.141.110.198) 162.950 ms r-vr3
  2-vl11.opb.interbusiness.it (217.141.110.199) 163.943 ms r-vr31-vl2.opb
    .interbusiness.it (217.141.110.138) 160.962 ms
3 r-mi213-vr32.opb.interbusiness.it (151.99.99.97) 171.940 ms r-mi208
  -vr31.opb.interbusiness.it (151.99.99.93) 171.607 ms r-mi213-vr32.opb.interbusiness.it
    (151.99.99.97) 167.950 ms
4 r-mi223-vl4.opb.interbusiness.it (151.99.75.214) 165.021 ms r-mi100
  -vl3.opb.interbusiness.it (151.99.75.136) 161.922 ms r-mi223-vl4.opb.interbusiness.it
    (151.99.75.214) 161.949 ms
5 mil8-ibs-resid-15-it.mil.seabone.net (195.22.205.157) 165.952 ms
  mil14-ibs-resid-8-it.mil.seabone.net (195.22.208.193) 163.949 ms mil8-ibs
    -resid-15-it.mil.seabone.net (195.22.205.157) 168.977 ms
6 mil25-mil7-racc1.mil.seabone.net (195.22.205.227) 167.938 ms 170.9 43 ms 166.959 ms
7 level3-1-us-mil25.mil.seabone.net (195.22.205.90) 166.961 ms 164.9 40 ms 159.959 ms
8 ae-0-54.mp2.Milan1.Level3.net (213.242.64.50) 166.955 ms 167.954 ms 166.941 ms
9 ae-1-0.bbr1.Frankfurt1.Level3.net (212.187.128.30) 196.950 ms 194.
  936 ms so-4-1-0.bbr2.Frankfurt1.Level3.net (4.68.128.185) 186.957 ms
10 ae-12-53.car2.Frankfurt1.Level3.net (4.68.118.80) 196.937 ms
  ae-22-56.car2.Frankfurt1.Level3.net (4.68.118.176) 188.928 ms
  ae-22-52.car2.Frankfurt1.Level3.net (4.68.118.48) 194.943 ms
11 62.67.38.226 (62.67.38.226) 181.969 ms 62.67.38.222 (62.67.38.222) 188.900 ms
  62.67.38.226 (62.67.38.226) 188.947 ms
12 * 72.14.238.121 (72.14.238.121) 223.989 ms 221.927 ms
13 72.14.232.208 (72.14.232.208) 226.943 ms 218.922 ms 222.936 ms
14 72.14.232.141 (72.14.232.141) 222.954 ms 220.936 ms 217.034 ms
15 72.14.233.79 (72.14.233.79) 206.879 ms 201.919 ms 213.958 ms
16 66.249.94.46 (66.249.94.46) 209.947 ms 214.924 ms 200.157 ms
17 66.249.93.104 (66.249.93.104) 206.736 ms 207.403 ms 205.466 ms
samiel@darkstar:~#

```

14.4.3 Il comando host

Il comando `host` serve a mappare i nomi in indirizzi IP e viceversa. Nel primo esempio, viene inserito il nome di un sito e l'output ne fornisce principalmente l'indirizzo IP.

```

root@darkstar:~# host www.slackware.com
www.slackware.com is an alias for slackware.com.
slackware.com has address 64.57.102.34
www.slackware.com is an alias for slackware.com.
www.slackware.com is an alias for slackware.com.
slackware.com mail is handled by 1 mail.slackware.com.

```

Nel secondo esempio si segue la procedura opposta: viene inserito un indirizzo IP e l'output ne fornisce il nome.


```
root@darkstar:~# host 64.57.102.34
34.102.57.64.in-addr.arpa domain name pointer slackware.com.
```

L'opzione più utile è `-a` (all), che indica di compiere una ricerca generale (cioè di tipo ANY).

```
root@darkstar:~# host -a www.slackware.com
Trying "www.slackware.com"
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 58509
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 2, ADDITIONAL: 0

;; QUESTION SECTION:
;www.slackware.com.          IN      ANY

;; ANSWER SECTION:
www.slackware.com.         75867  IN      CNAME   slackware.com.

;; AUTHORITY SECTION:
slackware.com.            75867  IN      NS      ns2.cwo.com.
slackware.com.            75867  IN      NS      ns1.cwo.com.

Received 89 bytes from 212.216.112.222#53 in 193 ms
```

14.4.4 Il comando dig

Il comando `dig` (acronimo di «Domain Information Grouper») è uno dei più potenti per rinvenire informazioni relative al DNS, compresi i lookup inversi. La sintassi è: `dig [server] [nome] [tipo]`. Se viene omesso il campo `[server]`, `dig` usa quello trovato in `/etc/resolv.conf`; il campo `[nome]` è il nome da risolvere; infine il campo `[tipo]` indica il tipo di ricerca da effettuare: ANY, A, MX, SIG, ecc.; se non viene precisato il tipo, `dig` effettua una ricerca per un record di tipo A. Ecco un esempio del suo output:

```
root@darkstar:~# dig www.slackware.com

; <<>> DiG 9.3.2 <<>> www.slackware.com
;; global options: printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 27386
;; flags: qr rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 2, ADDITIONAL: 2

;; QUESTION SECTION:
;www.slackware.com.          IN      A

;; ANSWER SECTION:
www.slackware.com.         86400  IN      CNAME   slackware.com.
slackware.com.            86400  IN      A       64.57.102.34
```

```
;; AUTHORITY SECTION:
slackware.com.      86400   IN      NS      ns1.cwo.com.
slackware.com.      86400   IN      NS      ns2.cwo.com.

;; ADDITIONAL SECTION:
ns1.cwo.com.        123473  IN      A       64.57.100.2
ns2.cwo.com.        123473  IN      A       64.57.100.3

;; Query time: 729 msec
;; SERVER: 212.216.112.222#53(212.216.112.222)
;; WHEN: Thu Jun 15 21:51:31 2006
;; MSG SIZE rcvd: 137
```

14.4.5 Il comando nslookup

Il comando `nslookup`, atto a interrogare i server dei nomi, è ormai deprecato, e in suo luogo possono essere usati i comandi `host` oppure `dig`. Nel primo esempio, viene inserito il nome di un sito e l'output ne fornisce principalmente l'indirizzo IP.

```
root@darkstar:~# nslookup www.slackware.com
Server:          212.216.112.222
Address:         212.216.112.222#53

Non-authoritative answer:
www.slackware.com canonical name = slackware.com.
Name:   slackware.com
Address: 64.57.102.34
```

Nel secondo esempio si segue la procedura opposta: viene inserito un indirizzo IP e l'output ne fornisce il nome.

```
root@darkstar:~# nslookup 64.57.102.34
Server:          212.216.112.222
Address:         212.216.112.222#53

Non-authoritative answer:
34.102.57.64.in-addr.arpa name = slackware.com.

Authoritative answers can be found from:
102.57.64.in-addr.arpa nameserver = ns1.cwo.com.
102.57.64.in-addr.arpa nameserver = ns2.cwo.com.
```

14.4.6 Il comando whois

In questo contesto, cioè avendo per argomento un nome di indirizzo, il comando `whois` visualizza alcune informazioni come nome e indirizzo del server e/o proprietario; inversamente, se l'argomento

è un indirizzo IP, `whois` visualizza il nome a cui appartiene. Le informazioni, se fornite, sono contenute in forma di record all'interno di un database di tipo RFC 3912. Di default, il comando cerca in `whois.networksolutions.com` per i record NIC e in `whois.arin.net` per gli indirizzi IPv4 e i nomi di rete. Ad esempio:

```
root@darkstar:~# whois www.google.com

Whois Server Version 2.0

Domain names in the .com and .net domains can now be registered
with many different competing registrars. Go to http://www.internic.net
for detailed information.

Server Name: WWW.GOOGLE.COM.TW
Registrar: ENOM, INC.
Whois Server: whois.enom.com
Referral URL: http://www.enom.com

Server Name: WWW.GOOGLE.COM.TR
Registrar: TUCOWS INC.
Whois Server: whois.opensrs.net
Referral URL: http://domainhelp.tucows.com

Server Name: WWW.GOOGLE.COM.PE
Registrar: ABACUS AMERICA, INC. DBA NAMES4EVER
Whois Server: whois.names4ever.com
Referral URL: http://www.names4ever.com

Server Name: WWW.GOOGLE.COM.MX
Registrar: ENOM, INC.
Whois Server: whois.enom.com
Referral URL: http://www.enom.com

Server Name: WWW.GOOGLE.COM.BR
Registrar: ENOM, INC.
Whois Server: whois.enom.com
Referral URL: http://www.enom.com

Server Name: WWW.GOOGLE.COM.AU
Registrar: MELBOURNE IT, LTD. D/B/A INTERNET NAMES WORLDWIDE
Whois Server: whois.melbourneit.com
Referral URL: http://www.melbourneit.com

Server Name: WWW.GOOGLE.COM.AR
Registrar: ENOM, INC.
Whois Server: whois.enom.com
Referral URL: http://www.enom.com
```

```
>>> Last update of whois database: Thu, 15 Jun 2006 15:55:35 EDT <<<
```

14.4.7 Il comando finger

Il comando `finger` serve a procurare informazioni su un determinato utente (come lo username, l'indirizzo, il numero di telefono, il sito di riferimento). Tali informazioni sono assunte da un qualche server `fingerpassword`, ma anche da dei file approntati a tal fine dall'utente stesso, come `~/plan`, `~/project` e `~/forward`. Quest'ultimo indica un indirizzo di posta elettronica a cui viene dirottata automaticamente la posta (e che può essere consultato anche da remoto: ecco la sua utilità).

Il servizio, in Slackware, è presente in `/etc/inetd.conf`, ma di default risulta disabilitato (mentre è presente la corrispondente porta in `/etc/services`).

14.4.8 Il comando telnet

Il comando `telnet` si utilizza per fare ingresso e svolgere determinate operazioni all'interno di un altro computer. È sufficiente digitare il comando seguito dal nome dell'host (che può essere un server `http`, `SMTP` o `POP3`): se questo permette la connessione, viene richiesta una password. Per uscire si usano i comandi `exit` o `logout`.

Telnet è tuttavia pericoloso o comunque insicuro, in quanto non cripta il flusso di informazioni, e quindi ne è fortemente sconsigliato l'uso attraverso Internet (in suo luogo, conviene in questo caso usare `ssh`).

14.4.9 Il comando ssh

Il servizio `ssh` consente, come Telnet, di connettersi a un computer remoto e di svolgere delle operazioni, ma esso cripta i dati trasmessi. Benché sia sempre possibile intercettare questo flusso, i dati catturati risulterebbero in ogni caso inutilizzabili.

14.5 Connettersi a Internet

14.5.1 Gli Winmodem

Com'è largamente noto, spesso i computer vengono (o almeno fino a qualche anno fa venivano) dotati di winmodem che, in realtà, non sono autentici modem, ma dispositivi in cui una parte fondamentale dei compiti è attribuita al software, cioè ai driver, che deve emulare una porta seriale. Poiché molti produttori non rilasciano driver per Linux per questo tipo di dispositivi e non ne rendono note le specifiche tecniche, essi non possono funzionare sotto Linux. Del resto, con un po' di attenzione al momento dell'acquisto, non è affatto difficile scartare questo tipo di hardware e scegliere un modem autentico. Un winmodem si riconosce perché in genere porta una delle seguenti diciture: HCF, HSP e HSF.

Per ovviare a questo limite, soprattutto negli anni in cui il modem 56kb era l'unico mezzo per effettuare una connessione a Internet, sono stati comunque sviluppati dei driver per Linux per parecchi winmodem. Il sito fondamentale in cui sono reperibili driver e informazioni in grande quantità è: <http://www.linmodems.org>. Dato il numero elevatissimo dei modelli supportati e le differenti caratteristiche dei driver, è impossibile fornire indicazioni generali sulla loro compilazione e installazione, che andrà valutata caso per caso.

Per conoscere le caratteristiche tecniche dei winmodem, è possibile effettuare una scansione del dispositivo via Internet, all'indirizzo: <http://132.68.73.235/linmodems/index.html#scanmodem> e scaricando l'utilità dedicata. Questa, una volta lanciata, crea una directory /modem con una serie di dati. Se scarsamente decifrabili, l'utente può fare ricorso alla mailing list discuss@linmodems.org per ottenere chiarimenti. Per conoscere il chip del modem, informazione altrettanto importante, si può fare una ricerca nel sito: <http://modemsite.com/56k/chipset.asp>.

14.5.2 Connessione con un modem esterno 56KB

Ipotizziamo di aver risolto gli eventuali problemi riguardanti il riconoscimento del nostro (win)modem interno, o di avere un modem interno o esterno che non è un winmodem. Innanzitutto possiamo creare un link simbolico dal dispositivo in nostro possesso a /dev/modem, o manualmente, o mediante un'utilità di `pkgtool`: è sufficiente andare alla voce **SETUP** e quindi **90.mode-dev**. Si sceglie la porta a cui è commesso il modem, generalmente `ttyS0`, che corrisponde in Windows alla porta COM1. Questo passaggio non è strettamente necessario, perché anche `pppsetup` crea lo stesso link automaticamente.

Impostare la connessione con `ppp-setup`

Per impostare la connessione utilizziamo l'utilità testuale `pppsetup`, che effettua la configurazione con alcuni semplici passaggi. In generale, è necessario aggiungere la stringa `OK` dopo ogni stringa di inizializzazione. Le stringhe contenenti il simbolo della e commerciale (&) devono essere poste fra virgolette doppie.

PHONE NUMBER: dobbiamo inserire il numero telefonico del nostro provider, cui va premessa la stringa di chiamata `atdt` se il telefono è a toni, `atdp` se il telefono è a impulsi. Ad esempio, sarà da inserire una stringa come: `atdt7020001033 ok`.

MODEM DEVICE: dobbiamo scegliere la porta a cui è connesso il modem. Come abbiamo visto, i modem su porta parallela normalmente corrispondono ai dispositivi `ttySn`.

BAUD RATE: si tratta della velocità del modem in Kbs (kilobyte per secondo). Per un modem 56KB, corrisponde al valore 57600.

CALLBACK: si tratta di una modalità di sicurezza che pochi provider oggi adottano. L'utente chiama il provider e questi, a sua volta, lo richiama per stabilire la connessione a Internet.

MODEM INIT STRING: si deve inserire la stringa di inizializzazione del modem. In linea di massima, per chiamare i numeri dei provider italiani è sufficiente la stringa `ATZ OK`. Si può impostare anche il volume del modem, scegliendo fra tre stringhe: `M0L0` (volume azzerato), `M1L1` (volume medio) e `M1L3` (volume alto).

I(nternet) S(ervice) P(rovider)'S DOMAIN NAME: va inserito il dominio del provider, ad esempio `tin.it`.

IP ADDRESS: va inserito l'indirizzo IP (numerico) del provider, ad esempio `212.216.112.222`.

PAP OR CHAP: va inserito il metodo di autenticazione adottato dal provider. In genere, i provider italiani adottano PAP oppure PAP/CHAP.

USERNAME: va inserito il nome dell'utente.

PASSWORD: va inserita la password relativa all'account in via di creazione. Questa viene scritta in chiaro nel file `/etc/ppp/pap-secrets`.

A questo punto una finestra ricapitola i dati inseriti (che sono riportati in `/etc/ppp/pppsetup.txt` per un'eventuale successiva lettura) e fornisce alcune informazioni sui servizi offerti da ppp. I dati della connessione sono stati scritti nel file `/etc/ppp/pppscript`, le opzioni di configurazione nel file `/etc/ppp/options`, le opzioni di connessione nel file `/etc/ppp/options.demand` e – come sappiamo – la password in `/etc/ppp/pap-secrets`. L'IP del provider viene infine scritto in `/etc/resolv.conf`.

A questo punto, tutto è pronto per lanciare la connessione col comando `ppp-go` e per interromperla col comando `ppp-off`.

Un piccolo inconveniente si genera se il computer è dotato di una scheda di rete, ovviamente non utilizzata per la connessione via modem. Infatti il routing è instradato di default verso il gateway registrato in `/etc/rc.d/rc.inet1.conf`, che potrebbe essere qualcosa come `192.168.0.1`, e di conseguenza la connessione ha luogo, ma non si riesce ad aprire alcun sito. Siamo avvertiti di questa situazione se, lanciando `ppp-go`, fra i dati dell'output leggiamo anche:

```
not replacing existing default route via 192.168.0.1
```

Per ovviare contingentemente a tale inconveniente è sufficiente reinstradare il traffico, in modo che passi non per il gateway ma direttamente per il modem, col comando: `route add default ppp0`. Possiamo invece ovviarvi stabilmente commentando in `/etc/rc.d/rc.inet1.conf` la riga:

```
# Default gateway IP address:
# GATEWAY="192.168.0.1"
```

Impostare la connessione con Kppp

Una comoda alternativa è costituita da **Kppp**, un'utilità di KDE da cui si può configurare, avviare e fermare la connessione (sotto Gnome ne esiste l'analogo, **Gnome PPP**). Essa modifica eventuali file di configurazione preesistenti (come se si fossero già creati con `ppp-setup`), ma vi lascia, commentandole, le vecchie entrate. Il vantaggio di questa utilità risiede nel fatto che, mentre `ppp-on` e `ppp-off` possono essere lanciati solo da root, **Kppp** funziona indifferentemente da root e da utente, anche se è sempre possibile modificare i permessi di quei comandi (cattiva soluzione), autorizzarli a un determinato gruppo o ancora autorizzarli soltanto per certi utenti con `sudo`.

Lanciando la configurazione con l'omonimo pulsante dalla finestra principale, bisogna ricorrere in ogni caso alla configurazione manuale, perché **Kppp** presenta valori preimpostati non validi per le connessioni italiane. I dati richiesti sono quasi tutti immediatamente intelligibili: nella scheda **Account** si deve creare un account nuovo con un nome a piacere; da qui si raggiunge la linguetta **Effettua chiamata**, dove bisogna inserire il nome del provider, il suo numero telefonico, scegliere l'autenticazione **PAP/CHAP**, e scegliere **Nessuno** come metodo di callback. Nella linguetta **IP** vale la pena di impostare ad **Automatico** l'indirizzo IP; nella linguetta **Gateway** si opta per **Gateway predefinito** e si mette la spunta alla voce **Assegna il "Default Route" a questo gateway**. Nella linguetta **DNS** si inserisce il nome di dominio del provider e si aggiungono l'indirizzo o gli indirizzi IP del suo server DNS. Un piccolo «trucco» per conoscere l'IP del DNS di un provider (posto che si sia già connessi a Internet) è lanciare dapprima il comando: `host -t NS dominio_del_provider` per conoscere il nome del server DNS e quindi: `host nome_del_server_DNS` per ottenerne l'indirizzo IP. Ad esempio, avremo:

```
samiel@darkstar:~$ host -t NS tin.it
tin.it name server dns.tin.it.
tin.it name server dnsca.tin.it.
```

Quindi, prendendo il secondo nome:

```
host dnsca.tin.it.  
samiel@darkstar:~$ host dnsca.tin.it  
dnsca.tin.it has address 195.31.190.31
```

Un'utile funzione attivabile all'ultima linguetta di questa sezione è **Costi telefonici**; basta selezionare il tipo di connessione cui facciamo ricorso per sapere in ogni istante a quanto ammonta il totale delle nostre spese. Per il resto, tutto può essere lasciato com'è.

Nella scheda **Modem** si aggiunge un modem nuovo con un nome identificativo a piacere, gli si attribuisce un dispositivo (che può essere `/dev/ttySn` oppure il link simbolico `/dev/modem`), si sceglie la velocità di connessione (che, per un modem 56 kbps, sarà 57600). In questa scheda, oltre al pulsante per interrogare il modem verificando se viene riconosciuto, bisogna prestare una certa attenzione alla scheda **Comandi modem**. Poiché le caratteristiche tecniche delle connessioni variano da Stato a Stato e in genere di default Kppp è impostato secondo parametri validi per gli Stati Uniti (dove il segnale di libero è costituito da un tono continuo, mentre in Italia esso è intermittente), alcune stringhe possono produrre dei fastidi. In Italia è allora possibile connettersi o lasciando disabilitata la voce **Attendi il segnale di linea prima di comporre il numero** o adeguando le stringhe che definiscono i comandi del modem: la stringa di inizializzazione sarà ATZ, la stringa di chiama ATDT se il telefono è a toni, ATDP se il telefono è a impulsi.

Infine, nella linguetta **Varie** mettendo la spunta alla voce **Aggiungi al pannello appena connesso** si riduce il programma a icona, ma controllando il tempo, la velocità ed eventualmente il costo della connessione attiva.

14.5.3 Connettersi a Internet con un modem ADSL

La connessione mediante la banda larga mette in campo concetti nuovi, da cui discendono le scelte pratiche sul tipo di hardware da adottare. Segnaliamo quanto meno due tipi di versi di protocolli (PPPoA e PpPoE) e tre tipi di hardware per le connessioni ADSL (modem USB, modem Ethernet e router).

Il **protocollo PPPoA** (Point to Point Protocol over ATM) è relativo a connessioni con modem ADSL di tipo USB.

Il **protocollo PpPoE** (Point to Point Protocol over Ethernet) è relativo a connessioni con modem ADSL collegati a porte Ethernet (in pratica, a schede di rete).

14.5.4 Connettersi con un modem USB

I modem USB sono dispositivi dotati a un estremo di un connettore per la presa telefonica (di tipo RJ-11) schermato da apposito filtro, all'altro di una presa USB da connettere al computer. Tali dispositivi richiedono dei driver per funzionare, cosicché ci si trova a volte in una situazione analoga a quella degli Winmodem. Per trovare i driver per GNU/Linux (ammesso e mai concesso che esistano) bisogna conoscere il chip del modem. Una volta in possesso di questo dato, si effettuerà una ricerca in siti come (citiamo i due di riferimento): <http://dsl.linux.it/DSLChipsets> e <http://www.adsl-hq.com/adslusbmodems>, ricchi di link ai progetti che portano avanti lo sviluppo di driver per GNU/Linux.

Anche qui, dato il numero elevatissimo dei modelli supportati e le differenti caratteristiche dei driver, è impossibile fornire indicazioni generali sulla loro compilazione e installazione, che andrà valutata caso per caso.

Menzioniamo però un problema diffuso con l'installazione di questo tipo di modem. Non è infrequente che hotplug lo riconosca erroneamente come una scheda audio, attribuendogli il modulo `dabusb`. Per rimediare, si deve «blacklistare» tale modulo, ossia inserirne il nome nel file `/etc/hotplug/blacklist` per impedirne in ogni caso il caricamento.

14.5.5 Connettersi con un modem Ethernet

I modem Ethernet sono simili ai modem ADSL, soltanto che a un estremo presentano un connettore (di tipo RJ-45) che va collegato alla scheda Ethernet del computer. Nell'ambito dei modem, questa è la scelta che garantisce la massima compatibilità con i sistemi GNU/Linux.

Per impostare la connessione, si ricorre in Slackware un'utilità testuale, `pppoe-setup`, a cui va fornita una serie di dati:

USER NAME: il nome utente nella forma `nome@dominio`.

INTERFACE: l'interfaccia di rete (di default `eth0`). In questo momento, è opportuno rispondere negativamente al quesito sull'attivazione del link on demand.

DNS: vanno indicati il DNS primario e (se presente) quello secondario del provider

PASSWORD: la password di connessione, da digitare due volte per sicurezza.

FIREWALLING: il livello di firewall che si intende attivare. Esso non va comunque attivato per un computer che funga da server (che richiede un firewall appositamente configurato), perché altrimenti le regole poste bloccherebbero l'accesso a tutti i server standard come server web, e-mail, ftp ecc. Se si usa SSH, tali regole bloccherebbero ogni connessione SSH in uscita che allocano una porta di uscita privilegiata. I livelli proposti sono tre:

0 – NONE = nessuna regola

1 – STANDALONE = regole generiche valide per un computer standing-alone

2 – MASQUERADE = regole generiche valide per un computer che opera come gateway in una LAN.

Si sceglie inserendo il numero corrispondente. Si tratta naturalmente di impostazioni generali, valide per la maggior parte dei casi.

Terminata la configurazione, viene mostrato un sommario e chiesto se si accettano queste impostazioni. Esse vengono registrate nei seguenti file:

in `/etc/ppp/pppoe.conf`: le impostazioni della connessione

in `/etc/resolv.conf`: gli indirizzi del server DNS (prima viene comunque effettuato un backup dell'eventualmente preesistente `/etc/resolv.conf` come `/etc/resolv.conf-bak`)

in `/etc/ppp/pap-secrets` e in `/etc/ppp/chap-secrets`: le password di autenticazione al server del provider (prima viene comunque effettuato un backup degli eventualmente preesistenti `/etc/ppp/pap-secrets` e `/etc/ppp/chap-secrets`, rispettivamente come `etc/ppp/pap-secrets-bak` e `/etc/ppp/chap-secrets-bak`).

A questo punto, possiamo gestire la connessione appena creata con una serie di comandi da shell:

`pppoe-start` = fa partire la connessione

`pppoe-stop` = fa cessare la connessione

`pppoe-status` = mostra alcune informazioni sulla connessione

`pppoe-connect nome_script` = permette di modificare uno script di connessione.

Ancora una volta, tutti questi comandi sono di proprietà di `root:root`. Per renderli accessibili agli utenti, valgono le medesime strategie che sono state indicate sopra a proposito della connessione mediante modem 56kb.

14.5.6 Connettersi a Internet con un router

La connessione mediante router implica la modalità più semplice di accesso e la garanzia che tutto funzionerà senza il benché minimo problema anche sui sistemi GNU/Linux, posto che si abbia una connessione flat, cioè senza limiti di tempo. Per vero, alcuni router, nella finestra di configurazione, danno modo di attivare e disattivare la connessione, come pure di impostare una disconnessione a tempo per inattività (in altre parole, se il router rileva che, in un certo lasso di tempo indicato dall'amministratore che lo imposta, non c'è alcun flusso di dati, effettua automaticamente la disconnessione). È infatti sufficiente che il computer sia dotato di una scheda Ethernet; il router, oltre a consentire in certi casi il collegamento di più computer (è il caso di quei router che hanno più porte e che fungono pertanto

anche da switch), integra spesso anche funzione di firewall, sia NAT (Network Address Translation) sia SPI (Stateful Packet Inspecting).

Il router va configurato via browser dal suo IP. Nella maggior parte dei casi, tale indirizzo è 192.168.0.1 (oppure 192.168.0.254); il manuale del router contiene comunque tutte queste indicazioni. Alcuni parametri, come il tipo di incapsulamento (PPPoA oppure PPPoE) e il metodo di multiplazione (LLC o VC), sono forniti dal provider. Non affrontiamo invece in questa sede la definizione di altri parametri, come l'impostazione di un firewall, poiché variano da marca a marca e da modello a modello. Certo, l'implementazione di un firewall a livello del router può rivelarsi una scelta molto felice.

Vi sono router che, come abbiamo detto, offrono un servizio DHCP, per cui è sufficiente collegarvi mediante un cavo la scheda o (se dotati di più porte e dunque in possesso della funzionalità aggiuntiva di switch) le schede di rete. In alternativa, si possono connettere allo stesso modo delle schede di rete già impostate con IP fisso. In ogni caso, il computer risulterà sempre connesso a Internet, senza necessitare di alcuna procedura per attivare o disattivare la connessione. Volendo, l'unico metodo è quello di spegnere fisicamente il router.

14.6 Impostare un firewall con Guarddog

14.6.1 Linux e il firewall

In GNU/Linux il firewall non è un'applicazione aggiuntiva, ma una funzionalità inserita nel kernel. Ciò lo rende particolarmente potente e versatile. Nelle versioni più recenti il firewall si vale di due componenti, **netfilter** che fa parte del kernel e che analizza le regole, e **iptables** (che rimpiazza il vecchio **ipchain**) che è l'interfaccia per comunicare con netfilter (per il quale a sua volta sono state create svariate interfacce grafiche). La configurazione manuale delle regole è tuttavia molto complessa, anche se consente di controllare una a una tutte le porte del computer: non la tratteremo in questa sede. In ogni caso, si possono verificare le porte aperte nel computer col comando `netstat -l`.

Per comodità, ci rivolgiamo a una comoda GUI che consente di configurare il firewall in modalità grafica: **Guarddog**, che si può scaricare all'indirizzo: <http://www.simonzone.com/software/guarddog> (un'altra è Shorewall, scaricabile all'indirizzo <http://www.shorewall.org>). Questa utilità crea anche il file `/etc/rc.firewall`, in cui sono conservate tutte le informazioni per il firewall. Se non trova una versione preesistente di questo file, produce un'avviso che però si può tranquillamente trascurare: il file sarà generato *ex novo*.

I servizi e le caselle da scegliere per consentire il traffico sono naturalmente relative ai servizi di cui ci si intende valere. Vediamo il dettaglio. Guarddog presenta quattro linguette: **Zona**, **Protocollo**, **Accesso** e **Avanzato**.

14.6.2 La linguetta Zona

Nella linguetta **Zona** si possono configurare gli indirizzi IP esterni, cioè quelli della navigazione in Internet, alla voce **Internet** e quelli interni, se ci troviamo all'interno di una rete, alla voce **Locale**: in altre parole, sia il traffico in uscita sia quello in entrata.

Possiamo creare una Nuova zona con un apposito pulsante, nel caso vogliamo attivare un determinato protocollo, perché il nostro computer lo possa utilizzare per connettersi a Internet, inserendo gli accessi (sotto forma di nomi di nodo, nomi di rete oppure indirizzi IP; le maschere di rete possono essere specificate con la notazione «dotted decimal» oppure con la notazione «slash») ad alcuni protocolli e IP specifici. Se, ad esempio, volessimo consentire al protocollo irc di connettersi solo a un determinato server, dovremmo creare una nuova zona, con un nome e un eventuale commento, inserire l'indirizzo del server, spuntare la casella **Locale** e quindi, nella linguetta **Protocolli** abilitare esclusivamente l'irc. A

questo punto il protocollo irc funzionerà solo verso quell'indirizzo. In tal modo è possibile mantenere aperte le porte del computer solo verso indirizzi che sappiamo essere affidabili.

Alla voce **Connessione** è possibile decidere se consentire il traffico da una zona a un'altra. Di default Guarddog consente il traffico da Internet a Locale e viceversa; ce ne rendiamo conto perché in Zona Internet è spuntata la casella Locale, mentre in Zona locale è spuntata la casella Internet.

14.6.3 La linguetta Protocollo

Nella linguetta **Protocollo** è possibile abilitare i vari protocolli a cui consentire il traffico e bloccare gli altri. Per comodità, Guarddog suddivide i protocolli in categorie.

La linguetta Zona Internet

Nella linguetta **Zona Internet** (traffico in uscita, dal nostro computer a Internet) dapprima possiamo abilitare in generale i servizi di cui ci serviamo e chiudere gli altri. I servizi inclusi sono:

Chat: se non si usano affatto chat, è inutile lasciare aperte le porte relative; se si usano solo alcune e non altre, scegliere opportunamente.

Gioco: va abilitato solo per chi pratica giochi online.

Media: serve solo a chi utilizza il plugin RealMedia.

Posta: è indispensabile permettere pop3 e smtp, in alcuni casi imap (altrimenti meglio lasciarlo escluso) per poter gestire la posta elettronica.

Rete: per un uso comune, si possono abilitare Dns e Ping; il resto serve per specifiche esigenze di gestione delle reti.

Servizio dati: possiamo abilitare i protocolli per la gestione di database come MySQL e PostgreSQL, se ne facciamo uso.

Sessione interattiva: si tratta di servizi particolari, che è necessario conoscere. Telnet è notoriamente insicuro, a differenza di SSH. Ma tutto dipende, ancora una volta, dei servizi di cui si ha bisogno.

Trasferimento file: sono indispensabili http (altrimenti non sarà possibile navigare in Internet) e ftp (per poter scaricare file da server ftp). Https ci serve per le connessioni a siti sicuri (ad esempio quelli di home banking o di commercio elettronico). In questa voce sono anche inclusi i servizi di file sharing e PeerToPeer.

Vari: è possibile abilitare qui il keyserver PGP per il controllo della posta elettronica e la cifratura dei messaggi.

La linguetta Zona locale

Nella **Zona locale** (traffico in entrata, da Internet al nostro computer) è necessario stabilire che cosa può passare dal nostro computer a Internet. In altre parole, **Zona locale** ci permette di definire quali servizi del nostro computer saranno accessibili dall'esterno, ed è perciò particolarmente delicata. Anche in questo caso, anzi a maggior ragione, è necessario limitare i servizi consentiti a quelli davvero essenziali. In particolare: alcuni servizi come Chat e Posta vanno abilitati simmetricamente a quanto fatto per la **Zona Internet**. Alla voce Rete va autorizzato soltanto Ident/Auth, indirizzo che molti server (fra cui ftp e POP3) interrogano quando ricevono la richiesta di una connessione. Se non si sa di che cosa esattamente si tratta, il resto va cautelativamente disattivato.

14.6.4 La linguetta Accesso

La terza linguetta **Accesso** determina i parametri del file di log, che registra tutti gli interventi del firewall. In questo caso si può accettare tranquillamente la configurazione proposta di default.

14.6.5 La linguetta Avanzato

Nell'ultima linguetta, **Avanzato**, oltre alla voce **Disabilita il firewall** (utile se dobbiamo riconfigurarla, magari a seguito di qualche errore grave di configurazione, ma lo stesso risultato si ottiene togliendo l'eseguibilità a `/etc/rc.firewall` con: `chmod a-x`), va segnalata la funzione **Nuovo Protocollo**, che serve a installare un protocollo non compreso nella lista della seconda linguetta. Nella sua creazione, gli viene assegnato un nome, il tipo di appartenenza (in genere TCP) e la porta, dopo di che lo troveremo nella linguetta **Protocolli**, alla voce **Definito dall'utente**, dove potremo attivarlo o disattivarlo e configurarlo come quelli già presenti di default.

Infine, se siamo connessi mediante una scheda di rete, dovremo abilitare il servizio DHCP relativamente al dispositivo con cui ci connettiamo.

Una volta terminata la configurazione, basta cliccare su **Applica** per salvare le modifiche apportate. Troveremo, già eseguibile, lo script `/etc/rc.firewall`. Il firewall può a questo punto essere testato su alcuni siti appositi, come <http://www.pcflank.com>.

Capitolo 15

Condivisioni con NFS

15.1 Connettere i computer

GNU/Linux è il sistema operativo più potente e versatile in fatto di reti; è infatti in grado di comunicare con una grandissima quantità di sistemi operativi diversi, a cominciare da Windows e dai sistemi Apple.

Ci occupiamo qui della condivisione di file in una rete interna (cioè in una LAN) fra computer che hanno tutti GNU/Linux come sistema operativo: in questo caso ci serviremo di NFS. C'è anche la possibilità di stabilire una condivisione fra computer che hanno GNU/Linux e altri che hanno Windows oppure OS/2: in quest'altro caso ci varremo di Samba.

Presupponiamo che i computer siano correttamente configurati per quanto attiene alla scheda di rete e che si vedano. Possiamo verificarlo col comando `ping`. Ipotizziamo di avere il `computer_1` con indirizzo IP 192.168.0.2 e il `computer_2` con indirizzo IP 192.168.0.3. Lanciando dal `computer_1` il comando: `ping -c1 192.168.0.3` (l'opzione `-c` specifica quanti pacchetti ICMP si inviano), dovremmo ricevere un output del tipo:

```
PING 192.168.0.3 (192.168.0.3) 56(84) bytes of data.  
64 bytes from 192.168.0.3: icmp_seq=1 ttl=64 time=0.371 ms  
--- 192.168.0.3 ping statistics ---  
1 packets transmitted, 1 received, 0% packet loss, time 0ms  
rtt min/avg/max/mdev = 0.371/0.371/0.371/0.000 ms
```

L'aspetto per noi rilevante è la presenza del messaggio «1 received», che ci mostra che il viaggio del pacchetto di prova è andato a buon fine. Ripetiamo simmetricamente l'operazione dal secondo computer. Se invece si ricevesse «0 received», allora bisognerebbe ricontrollare la rete: o l'hardware, o l'assegnazione degli indirizzi IP effettuata con `netconfig` (essi sono riscontrabili in `/etc/rc.d/rc.inet1.conf`) dovrebbero contenere degli errori.

15.2 Il protocollo NFS

Il protocollo NFS (acronimo di «Network File System») consente di «comunicare» a due o più computer che hanno un sistema operativo GNU/Linux. Slackware è già predisposta per questo servizio; possiamo comunque compiere una verifica col comando: `cat /proc/filesystems`, che fra le altre righe dovrà darcene una come la seguente:

```
nodev nfs
```

Il computer che mette i suoi dati a disposizione si chiama *server*, quello che li riceve si chiama *client*. Poiché la condivisione è gestita sulla base dell'indirizzo IP dei vari computer, la rete dev'essere settata con indirizzo IP statico, e non con l'indirizzo IP dinamico stabilito da DHCP. In realtà, sarebbe possibile utilizzare DHCP con indirizzi dinamici, ma ciò presuppone che uno dei computer sia impostato come server DSN per la rete interna, e qui prescindiamo da questo più complesso scenario.

I demoni che gestiscono i vari aspetti della condivisione sono `/sbin/rpc.portmap`, `/usr/sbin/rpc.nfsd` e `/usr/sbin/rpc.mountd`: **rpc.portmap** fa sì che alle richieste NFS risponda il demone corretto, **rpc.mountd** serve all'effettuazione del montaggio del file system e **rpc.nfsd** risponde alle richieste del client, in base alle direttive di `/etc/exports`.

Se le impostazioni sono corrette, il comando `rpcinfo -p` ci restituirà un output del tipo:

```

programma  vers  proto  porta
100000      2     tcp    111    portmapper
100000      2     udp    111    portmapper
100011      1     udp    941    rquotad
100011      2     udp    941    rquotad
100011      1     tcp    944    rquotad
100011      2     tcp    944    rquotad
100003      2     udp    2049   nfs
100003      3     udp    2049   nfs
100003      4     udp    2049   nfs
100021      1     udp    32770  nlockmgr
100021      3     udp    32770  nlockmgr
100021      4     udp    32770  nlockmgr
100005      1     udp    951    mountd
100005      1     tcp    954    mountd
100005      2     udp    951    mountd
100005      2     tcp    954    mountd
100005      3     udp    951    mountd
100005      3     tcp    954    mountd
100024      1     udp    959    status
100024      1     tcp    962    status

```

La condivisione implica che nel server sia indicata la o le directory da condividere, e che nel client ne sia indicato il punto di montaggio. La configurazione di base è estremamente semplice e veloce, anche se può essere ulteriormente affinata.

15.3 Configurazione del server

Nel file `/etc/exports` indicheremo:

1. la (o le) directory da condividere
2. il (o i) computer con cui condividerla (o condividerle), ossia il (o i) client
3. i permessi, racchiusi fra parentesi tonde e senza lasciare alcuno spazio con l'indicazione del client.

1. si indica col percorso completo; la barra (/) indica che si intende montare l'intero file system. 2. si indica con l'indirizzo IP oppure col nome del computer client. Si avrà pertanto una riga come:

```
/ 192.168.0.2(rw,nohide,no_root_squash)
```

È possibile condividere soltanto una determinata directory, e allora la riga risulterà ad esempio:

```
/home/samiel 192.168.0.2(rw,nohide,no_root_squash)
```

È possibile specificare opzioni diverse per client diversi sulla stessa condivisione, ad esempio:

```
/home/samiel 192.168.0.3(rw,no_root_squash) 192.168.0.4(ro,root_squash)
```

In questo modo si condivide con due diversi client la stessa directory, ma, fra l'altro, si assegna al primo il permesso di lettura e scrittura, al secondo solo quello di lettura.

È possibile inoltre differenziare le directory e i permessi fra vari client. Ad esempio:

```
/ 192.168.0.4(ro,nohide,_netdev,no_root_squash)
/home/samiel 192.168.0.4(rw,nohide,_netdev,no_root_squash)
```

In questo modo il server mette a disposizione del primo computer il suo intero file system in sola lettura, e del secondo computer esclusivamente la home dell'utente samiel sia in lettura sia in scrittura.

Si possono inoltre usare dei caratteri jolly. Indicare come client *.mydomain.org significa concedere i permessi a tutti i computer che fanno parte di quel dominio.

Vediamo infine le opzioni principali:

ro il file system è disponibile in sola lettura (read only).

rw il file system è disponibile in lettura e scrittura (read & write).

sync sincronizza i dati fra server e client in modo che essi siano scritti su un supporto non volatile (come un disco fisso) prima di ogni operazione di condivisione. È l'opzione di default.

async autorizza la violazione del protocollo NFS, che risponde alle richieste prima che i dati siano scritti su di un supporto non volatile. In altre parole, privilegia il server rispetto al client. Questa opzione, che dev'essere precisata esplicitamente, velocizza il trasferimento dei dati, ma ne può determinare la corruzione.

nohide se il server esporta due file system di cui uno è montato sull'altro, il client deve montarli entrambi in modo esplicito per accedere loro. Se infatti viene montata solo la directory madre, la directory figlia appare vuota («nascosta»). Con questa opzione invece è possibile vedere entrambe le directory.

root_squash tratta l'utente root del client come utente nobody, il che significa che impedisce all'utente root di accedere al file system con i suoi privilegi. Si tratta di un'opzione di sicurezza atta a prevenire che tale utente possa modificare i file di sistema del server, e in effetti è il default.

no_root_squash disabilita l'opzione **root_squash**. Ha un'utilità effettiva soltanto nei client NFS privi di disco fisso.

Dopo aver modificato il file /etc/exports, le modifiche vengono applicate lanciando il comando: `exportfs -r`.

15.4 Configurazione del client

Si tratta adesso di montare stabilmente la condivisione editando `/etc/fstab`, per indicare:

1. l'indirizzo del server
2. la directory condivisa
3. il punto di montaggio
4. il file system, nel nostro caso sempre `nfs`
5. le opzioni.

1. e 2. sono scritte di seguito, separate solo dai due punti. Avremo perciò, ad esempio:

```
192.168.0.3:/ /mnt/brightstar nfs noauto,_netdev,sync,users 0 0
```

oppure:

```
192.168.0.3:/home/samiel /mnt/brightstar nfs noauto,_netdev,sync,users 0 0
```

Le opzioni più importanti sono:

rsize indica il buffer in lettura (`r = read`) in kb. Attualmente il default è 4096, che può essere aumentato a 8192.

wsize indica il buffer in scrittura (`w = write`) in kb. Attualmente il default è 4096, che può essere aumentato a 8192.

auto monta automaticamente il file system.

noauto non monta automaticamente il file system.

_netdev indica che il file system risiede su un dispositivo che richiede un accesso dalla rete. Serve a prevenire il tentativo di montaggio dell'unità finché la rete non sia stata abilitata.

sync consente la trasmissione sincrona dei dati di I/O. È l'opzione di default.

async consente la trasmissione asincrona dei dati di I/O.

hard Stabilisce che la connessione deve essere ritentata all'infinito, anche dopo un crash del sistema. È la modalità di funzionamento predefinita. Il tentativo può essere interrotto solo se si precisa `in.`

in consente di interrompere una chiamata NFS mediante dei segnali. È utile quando il server non risponde.

soft consente un timeout in caso di crash del server. Il timeout è precisato con `timeo`.

timeo definisce il valore del timeout, espresso in decimi di secondo, per il completamento delle richieste; il valore di default è 0,7 secondi. Si definisce con la sintassi `timeo=n`, dove *n* sono i decimi di secondo. Se in quel lasso di tempo non si ottiene una conferma, la connessione viene ritentata con una durata di timeout doppia, fino a un massimo di 60 secondi.

15.5 La risoluzione dei nomi

È possibile usare, in luogo dell'indirizzo IP dei computer, i loro nomi, a patto che sia attivato il servizio DNS, cioè la risoluzione dei nomi negli indirizzi. Ciò appare comodo soprattutto se i computer da gestire sono svariati e sarebbe impossibile ricordarne correttamente tutti gli indirizzi IP.

Come già sappiamo, per ottenere questo risultato, si edita il file `/etc/hosts` in tutti i computer (il server e i client) e si aggiungono, per ogni computer coinvolto nella condivisione:

1. l'indirizzo IP
2. il nome
3. eventualmente un alias, per abbreviare la configurazione.

Avremo pertanto delle righe come:

```
# For loopbacking.  
127.0.0.1 localhost  
192.168.0.2 darkstar.mydomain.org darkstar  
192.168.0.3 brightstar.mydomain.org brightstar  
# End of hosts.
```

In questo modo, invece di:

```
/ 192.168.0.2(rw,nohide,_netdev,no_root_squash)
```

potremo scrivere:

```
/ darkstar(rw,nohide,_netdev,no_root_squash)
```

15.6 Messaggi di errore

Se il montaggio del file system remoto non riesce, e si riceve il messaggio: «mount: RPC: Program not registered. Please check that the disk is entered correctly.», allora o il portmapper non sta girando, oppure qualcosa impedisce l'accesso al file system, verosimilmente in `/etc/hosts.allow` o in `/etc/hosts.deny`.

Se si riceve l'avviso «mount /: /mnt/brightstar failed, reason given by server: Permission denied», allora o c'è qualche errore in `/etc/exports`, oppure, se il file è stato modificato, non è stato lanciato il comando `exports -r`.

15.7 Il controllo degli accessi

Per garantire un accesso controllato, i permessi possono essere definiti nel server in modo particolareggiato nei file `/etc/hosts.allow` e `/etc/hosts.deny`. La strategia usata di solito consiste nel negare in `/etc/hosts.deny` tutti i permessi, specificandoli individualmente, come segue:

```
portmap:ALL  
lockd:ALL  
mountd:ALL  
rquotad:ALL  
statd:ALL
```

oppure complessivamente, come segue:

```
ALL: ALL
```

e nello stabilire poi determinati permessi a determinati client all'interno del file `/etc/hosts.allow`, come segue:

```
portmap:192.168.0.3 , 192.168.0.4
lockd:192.168.0.3 , 192.168.0.4
mountd:192.168.0.3 , 192.168.0.4
rquotad:192.168.0.3 , 192.168.0.4
statd:192.168.0.3 , 192.168.0.4
```

In questi due file vanno indicati gli indirizzi IP e non i nomi dei computer o i loro alias. Per accertarsi che portmap legga i due file in questione, si lancia il comando: `strings /sbin/rpc.portmap | grep hosts`.

15.8 Problemi di sicurezza

Il protocollo NFS trasmette i dati in chiaro, e pertanto presenta evidenti vulnerabilità sotto il profilo della sicurezza. Ciò non pone grossi problemi nell'ipotesi, che qui abbiamo fatta nostra, di una rete domestica o comunque sicura. Ci siamo infatti limitati a controllare gli accessi, anche se sarebbe stato possibile implementare un firewall che controllasse non solo l'indirizzo IP del client che accede alle risorse condivise, ma anche il suo MAC address.

È altamente sconsigliato usare NFS per esportare dati in rete; in questo caso si possono prendere comunque delle misure precauzionali, a cui accenneremo soltanto. Innanzitutto l'esportazione va fatta in modalità di sola lettura; in secondo luogo va limitata l'eseguibilità dei binari. Infine, data l'insicurezza intrinseca di RPC, si deve ricorrere a un metodo che risolva il problema della trasmissione dei dati in chiaro. RPC (acronimo di «Remote Procedure Calling») è un meccanismo di relazione client/server in base al quale il client trasmette al server la richiesta di eseguire un processo o un'applicazione, mentre il server è in ascolto. Ricevuta la richiesta, il server comunica al client la porta TCP/IP da usare per stabilire la connessione al servizio richiesto. In questo caso una soluzione consiste nell'incapsulare NFS in SSH, ossia nel ricorrere al tunnelling di NFS via SSH, per cui i dati trasmessi verranno ora criptati.

Capitolo 16

Condivisioni con Samba

16.1 Che cos'è Samba

Samba, grazie al protocollo SMB (Server Message Block, noto anche come LAN Manager) consente la comunicazione in una rete «mista», tra computer GNU/Linux e computer Windows od OS/2 e la condivisione di dati e dispositivi (stampanti). Più precisamente, Samba nella sua ultima versione fornisce almeno i seguenti servizi:

1. server per condividere file system e stampanti
2. client per accedere a risorse NetBIOS su macchine Unix, Windows e OS/2
3. master browser sia locale sia di dominio
4. server WINS (acronimo di «Windows Internet Name Service»). Il ricorso a questa funzionalità è deprecato nelle ultime versioni di Samba, perché fornisce un supporto di base per la risoluzione dei nomi per macchine win 9x e Me, laddove ora Samba implementa un proprio servizio DNS dinamico
5. server per l'autenticazione di client in un dominio MS-Windows
6. server DFS (acronimo di «Distributed File System»)
7. server o client di Active Directory con supporto LDAP o Kerberos5 (la nuova implementazione di Samba quale controllore di dominio include un server LDAP built-in chiamato LDB e un'interfaccia per la distribuzione delle chiavi KDC, acronimo di «Kerberos Distribution Center»)
8. instaurazione di relazioni di fiducia con server MS-Windows-NT
9. strumenti per la migrazione rapida e automatica di un dominio gestito con PDC (Primary domain controller) e BDC (Backup domain controller) MS-Windows NT in un dominio gestito con PDC e BDC Samba (via un'interfaccia *web-based*)
10. introduzione di nuovo file system virtuale col supporto delle ACLs (Access Control Lists)
11. come Windows include Visual Basic e WSH (Windows Scripting Host), ora Samba include a sua volta una nuova interfaccia di scripting in JavaScript
12. un'implementazione – per il momento minima – del protocollo SMB2, presente in Windows Vista®

Le reti Unix in generale usano il protocollo TCP/IP, mentre Windows usa per l'appunto SMB sui protocolli IPX/SPX, NetBEUI e TCP/IP. Per consentire la comunicazione, Samba (sviluppato da Andrew Tridgell) adotta un'interfaccia di rete chiamata NetBIOS (acronimo di «Network Basic Input Output System») che consente ai due sistemi di dialogare. Dal canto suo, Microsoft ha sviluppato, allo stesso fine, il sistema CIFS (acronimo di «Common Internet File System»), ma Samba è più potente e veloce.

Samba è un programma complesso; mentre altre distribuzioni adottano una suddivisione dei suoi file in più pacchetti, Slackware li include tutti in uno unico, contenente i demoni, i programmi di servizio, le utilità e le relative librerie (nonché l'amplissima documentazione).

Il file base di configurazione di Samba, `/etc/samba/smb.conf`, è assai esteso e ricco di opzioni, prevedendone oltre trecento. In realtà, assai poche di queste servono per creare una piccola rete mista. Qui pertanto non ci occuperemo, fra l'altro, di problematiche come i domini, la gestione del browsing, il ricorso a LDAP e svilupperemo soltanto le seguenti quattro semplici problematiche: come condividere file in Windows con Slackware, come condividere file in Slackware con Windows, come stampare da Slackware con una stampante installata su Windows, come stampare da Windows con una stampante installata su Slackware, il tutto in una piccola rete sicura. La presente sezione si basa essenzialmente su Windows XP Pro SP2, poiché alcune versioni precedenti di Windows avevano un trattamento diverso delle password.

16.2 Come funziona Samba

Samba, in base al sistema di init tipico di Slackware, si basa sullo script `/etc/rc.d/rc.samba`, che avvia il servizio, e su due demoni che lo gestiscono, `/usr/sbin/smbd` e `/usr/sbin/nmbd`, e in più (eventualmente) sul demone `/usr/sbin/winbindd`.

Slackware, optando per la soluzione migliore, avvia i due primi due demoni come indipendenti, e cioè non gestiti da `/etc/inetd.conf`. Se infatti il loro avvio dipendesse da quest'ultimo file, essi erediterebbero i suoi privilegi, ma anche i suoi difetti; inoltre, se si dovesse fermare o riavviare Samba, sarebbe necessario fermare e riavviare anche tutti i servizi che dipendono da `/etc/inetd.conf`.

Il demone **smbd** fornisce i servizi di accesso e condivisione a file e stampanti ai client Samba e ad altri sistemi, autentica gli utenti, gestisce gli accessi e controlla le risorse condivise. Può essere avviato a mano, in questo caso con alcuni parametri. Fra essi, l'opzione `-s` consente di precisare una posizione diversa per il file di configurazione, e l'opzione `-l` consente di specificare un file di log. L'opzione `-D` con cui viene lanciato di default dal sistema lo fa partire appunto come demone. Le porte di default attraverso le quali il server esegue l'ascolto del traffico SMB sono le porte 139 e 445. Supporta una serie di comandi che fanno a capo allo script `/etc/rc.d/rc.samba`:

```
rc.samba start: avvia il servizio Samba
rc.samba stop: termina il servizio Samba
rc.samba restart: esegue in successione stop e start, leggendo nuovamente i file di configurazione
rc.samba reload: rilegge i file di configurazione
rc.samba status: visualizza lo stato dei demoni.
```

Il demone **nmbd**, che in realtà non è che un semplice name server, fornisce la risoluzione dei nomi NetBIOS; esso resta in ascolto delle richieste per il name server e risponde con le informazioni in suo possesso. Inoltre gestisce una lista delle risorse di rete per il browser e partecipa alla scelta del browsing. Anch'esso ha diverse opzioni, atte soprattutto a scopi di test e debug. L'opzione `-D` con cui viene lanciato di default dal sistema lo fa partire appunto come demone. Le porte di default attraverso la quale il server ascolta il traffico SMB sono le porte 137 e 138.

Il demone **winbind** è un servizio di tipo client-side che risolve le informazioni riguardanti l'utente e il gruppo su di un server Windows NT, rendendole comprensibili dalle piattaforme UNIX mediante le chiamate Microsoft RPC (Remote Procedure Call), Pluggable Authentication Modules (PAM) e Name Service Switch (NSS). Ciò permette agli utenti del dominio Windows NT di comportarsi come utenti

UNIX su di una macchina UNIX. Il servizio winbind è controllato separatamente dal servizio Samba. Qui non ci occuperemo di questa problematica.

Inoltre, il funzionamento di Samba fa riferimento a `/etc/services` e a `/etc/hosts`. Slackware è già pronta per tutto questo, per cui l'avvio del servizio non richiede che si metta mano a nessun file, tranne che a `/etc/hosts`, per fruire del servizio di risoluzione dei nomi. È sufficiente rendere eseguibile col comando `chmod u+x` il file `/etc/rc.d/rc.samba` (il comando `chmod a+x`, consentendo a tutti gli utenti di eseguire il file, potrebbe compromettere la sicurezza), oppure con l'utilità `pkgtool`, alla voce **Setup**, scegliendo **services** e quindi **rc.samba**.

Infine, il file preconfigurato `/etc/samba/smb.conf-sample` va rinominato `/etc/samba/smb.conf` per introdurre le modifiche necessarie (usarlo come punto di partenza è una buona idea, soprattutto quando si è alle prime configurazioni, per non perdere tempo ed essere certi di non tralasciare sezioni importanti), oppure crearne uno nuovo da zero.

16.3 Come predisporre Windows per Samba

Per configurare Windows, si presuppone che i computer sulla rete si vedano, cioè che sia possibile effettuare un'operazione di ping. Le operazioni di configurazione di Windows si riducono a poche, per modificare quei settaggi che impedirebbero a Samba di funzionare.

In questa sede presupponiamo – come indicato subito di seguito – che in tutti i computer l'indirizzo IP delle schede di rete sia impostato in maniera statica: si tratta infatti della soluzione più semplice per una LAN domestica. La situazione è differente se l'attribuzione degli indirizzi all'interno della rete viene compiuta da un server DHCP: in questo caso tali indirizzi potrebbero variare ogni volta, e comunque sarà impossibile utilizzare il nome degli host, a meno di non ricorrere a un name server. Qui però non trattiamo questo più complesso scenario.

In primo luogo, la scheda di rete dei computer Windows, che di default riceve un indirizzo IP dinamico mediante DHCP, va impostata con un IP fisso (questo per evitare la complicazione ulteriore di dover impostare un server DHCP/DNS, cosa peraltro affatto praticabile). Per far ciò, si apre **Impostazioni – Pannello di controllo – Connessioni di rete – LAN o Internet ad alta velocità – Connessioni alla rete locale (LAN)** e si clicca col pulsante destro del mouse sulla scheda di rete pertinente. Nella finestra che appare, dal titolo **Proprietà - Connessione alla rete locale (LAN)**, nella prima linguetta, **Generale**, si seleziona la voce **Protocollo Internet (TCP/IP)** e si preme il tasto **Proprietà**. Si mette la spunta a **Utilizza il seguente indirizzo IP**. Ponendo di essere all'interno di una rete di classe C, andranno inseriti l'indirizzo IP fisso della scheda di rete (qualcosa come 192.168.0.3), la Subnet mask (255.255.255.0) e il Gateway predefinito (l'indirizzo del router, pertanto 192.168.0.1). Nelle righe successive andranno inseriti il DNS primario e secondario della connessione (forniti dal provider). Le altre voci possono essere lasciate come predefinite.

In secondo luogo, in **Impostazioni avanzate TCP/IP** si va alla linguetta **WINS**. Alla voce **Indirizzi WINS** in ordine di utilizzo si preme il bottone **Aggiungi** e si inserisce l'IP del server Samba. Sotto, in **Impostazioni NetBIOS**, poiché abbiamo attribuito un indirizzo statico alla scheda di rete, selezioniamo la voce **Abilita NetBIOS su TCP/IP**.

Infine, da **Pannello di controllo – Sistema**, nella scheda **Proprietà del sistema** si sceglie la linguetta **Nome computer** e si preme il pulsante **Cambia**. Appare una finestra denominata **Cambiamenti nome computer**. Si attribuisce alla macchina un nome identificativo e, poiché abbiamo escluso di occuparci dei domini, mettiamo la spunta alla voce **Gruppo di lavoro**. Denominiamo tale gruppo, ricordando che, di default, in Windows è `workgroup`; vale qui qualsiasi nome, che dovrà essere indicato anche in `/etc/samba/smb.conf`.

Non bisogna dimenticare che i firewall di Windows impediscono a Samba l'accesso: vanno pertanto impostate delle regole specifiche che lo consentano, regole che assumono forma diversa a seconda del firewall adottato.

16.4 Il file di configurazione `/etc/samba/smb.conf`

16.4.1 Aspetti generali della configurazione

Che cosa e come si condivide, dipende da `/etc/samba/smb.conf`, il file di configurazione di Samba. Il file di default `/etc/samba/smb.conf-sample` è documentato molto bene (ogni sezione e sottosezione è infatti commentata in modo ampio e chiaro), sebbene non risolva le problematiche più complesse, ad esempio LDAP (su questo tema può innanzi tutto vedersi il file di configurazione di esempio `/usr/doc/samba-numero_versione/examples/LDAP/samba.schema`), Active Directory e le numerose implementazioni del controller del dominio. Dato che le dimensioni che può raggiungere sono considerevoli, tale file è strutturato in diverse sezioni in certi casi dotate, a loro volta, di sottosezioni.

Questo file presenta una sua sintassi, che dev'essere tenuta presente:

- le righe da commentare sono precedute da cancelletto (#) o punto e virgola (;). Per convenzione, quanto è preceduto da # è in genere un testo esplicativo, mentre quanto è preceduto da ; è un comando o una riga di configurazione disabilitata.
- le stringhe di configurazione possono estendersi su più righe utilizzando una barra inversa (\) a fine riga.
- le opzioni e i valori sono «case insensitive», ma se si indica un path nel file system questo è «case sensitive».
- per separare una serie di valori possono essere usati sia virgole (,) che spazi vuoti.
- possono essere utilizzate delle variabili (di ambiente), precedute dal simbolo di percentuale (%) all'interno dei valori (ad esempio: `path = /home/%u`). Il nome di queste variabili è sostituito dal corrispettivo valore quando i demoni lo leggono:

%a	architettura software della macchina remota (Samba, MS-Windows 95/98/Me/NT); diversamente sarà assegnato il valore UNKNOWN
%d	numero identificativo del processo corrente
%D	nome del dominio MS-Windows in cui il server Samba si integra
%g	nome del gruppo primario di %u
%G	nome del gruppo primario di %U
%h	nome del nodo che ha avviato il servizio Samba
%H	directory personale assegnata a %u
%I	indirizzo IP della macchina client
%L	nome NetBIOS assegnato al server
%m	nome NetBIOS della macchina client
%M	nome Internet della macchina client
%N	nome della directory personale ottenuta dal servizio NIS del server
%p	percorso della directory personale ottenuto dal servizio NIS
%P	directory principale del servizio corrente
%S	nome della condivisione corrente ([tmp], [homes], ecc.)
%T	data e ora corrente
%u	nome dell'utente (GNU/Linux) del servizio corrente
%U	nominativo dell'utente della sessione
%v	versione in uso di Samba

- si può includere in `/etc/samba/smb.conf` un altro file di configurazione con l'opzione `include` e il percorso assoluto del file (ad esempio `include = /etc/samba/smb.conf.%a`).

- le variazioni apportate a questo file non richiedono il riavvio dei demoni di Samba, in quanto esso viene riletto di default ogni sessanta secondi. In ogni caso, le variazioni alla configurazione delle risorse condivise non vengono applicate finché c'è un qualche utente che vi è connesso.

Le principali sezioni di /etc/samba/smb.conf sono:

[global]: definisce le impostazioni di configurazione globali. Essa può contenere, a sua volta, le direttive [global] e [share]. Le prime definiscono le caratteristiche complessive e il comportamento generale di Samba; le seconde impostano il comportamento di default delle condivisioni.

[share definitions]: specifica le risorse condivise e il loro comportamento.

Le principali sottosezioni di [share definitions] sono:

[homes] specifica la condivisione delle directory home degli utenti e i loro percorsi e permessi

[printers] gestisce globalmente le stampanti condivise, definendo i relativi accessi

[tmp] definisce lo spazio da dedicare allo swap dei file temporanei

[public] definisce l'accesso a una specifica directory pubblica, cioè accessibile a tutti gli utenti

[share] indica particolari risorse condivise. Le si può attribuire un nome a piacere, purché sempre racchiuso fra parentesi quadre. Essa, a sua volta suddivisa in ulteriori sottosezioni, definisce la condivisione per risorse particolari, come alcune directory e soltanto quelle, e per gruppi appositamente definiti; una directory può essere infatti messa a disposizione di un controllato numero di utenti raccolti in un gruppo, oppure si può autorizzare soltanto un gruppo a scrivere sulla risorsa, e via dicendo.

Forniremo poco sotto alcuni esempi di simili impostazioni.

I fondamentali parametri che si possono impostare sono:

1. la definizione delle risorse da condividere
2. la definizione degli host che possono accedere a una risorsa condivisa
3. la creazione di eventuali password da richiedere gli utenti per accedere alle risorse
4. la modifica dei permessi agli utenti per le risorse condivise
5. i limiti della fascia oraria o del periodo in cui le risorse possono essere condivise.

Ogni parametro impostato nelle sezioni particolari prevale su quello specificato nella sezione [global]. Ogni sezione dev'essere aperta dal titolo, indicato fra parentesi quadre. Come sappiamo, le righe precedute dal simbolo del cancelletto (#) o del punto e virgola (;) sono commenti e perciò non vengono interpretati dal servizio.

16.4.2 La sezione [global]

Qui sono presenti i parametri che determinano la configurazione di Samba nel suo complesso. Se un parametro non è impostato nelle sezioni specifiche, Samba applicherà quelli che legge in questa sezione. Inoltre, è qui che si possono stabilire le impostazioni generali per la sicurezza. Questa sezione è l'unica di cui non è obbligatorio inserire il titolo; infatti Samba prende tutto quanto precede la prima sezione intitolata come appartenente alla sezione [global], ne sia precisato o meno il titolo.

Riportiamo le voci di questa sezione che è più utile conoscere.

Configurazione di base NetBIOS

`workgroup` corrisponde al «Gruppo di lavoro» di Windows, che di default è `WORKGROUP` e assegna a Samba il dominio o gruppo di appartenenza.

`netbios name` corrisponde al nome che viene attribuito al computer e serve ad assegnare al server un nome NetBIOS che è uguale a quello assegnato dal DNS senza il dominio. Se, ad esempio, il nome DNS del server fosse `darkstar.mydomain.org`, il nome NetBIOS sarebbe `darkstar`. Uno dei casi in cui si richiede questo campo è quello in cui la rete fosse suddivisa in due o più domini DNS diversi, perché potrebbero esserci in due sottoreti due macchine con lo stesso nome. I nomi NetBIOS consistono in un'unica stringa senza punti, contenente lettere maiuscole, minuscole, cifre e i simboli `!, @, #, $, %, ^, &, (,), -, ' e ~`. Questa voce può comunque essere tralasciata in una rete priva di sottoreti (e a maggior ragione in una LAN domestica).

`server string` la stringa che si legge in Windows alla voce «Descrizione del pc condiviso»: si tratta di una descrizione che può essere inserita a piacere. Di default viene indicata la versione corrente di Samba.

Gestione della sicurezza

La fondamentale voce `[security mode]` specifica il livello di sicurezza con cui vengono gestiti gli accessi. Le possibilità sono cinque:

`security = share` il controllo viene eseguito a livello di condivisione. Gli utenti che si connettono non vengono controllati, tanto che per la connessione non c'è bisogno di alcuna password. Samba procede tentando innanzitutto di dedurre il nome dell'utente dalla direttiva `valid users`, se presente nella sezione di condivisione della risorsa; secondariamente dal nome del client, o, in caso di insuccesso, da quanto indicato con il parametro `guest account`, se fra i parametri di condivisione è indicato `guest ok = yes` e `guest only = yes`. Questo livello di sicurezza si usa soprattutto nel caso di utenti MS-Windows e GNU/Linux non coincidenti, per condividere porzioni di file system quando si vuole tutti gli utenti abbiano gli stessi diritti sui file condivisi: infatti l'impostazione `guest account = utente_samba` fa sì che il proprietario delle risorse condivise sia sempre l'utente GNU/Linux `utente_samba` qualunque sia il nominativo dell'utente MS-Windows che si connette. È il livello maggiormente permissivo, adeguato solo a reti chiuse e sicure come può essere una piccola rete domestica, che, fra l'altro, esime dal dover digitare una password in accesso alle risorse. In questo caso, sebbene non serva, si può inserire l'opzione `null password = yes`, che precisa in forma esplicita che i client possono accedere alle risorse condivise senza alcuna password. Aggiungiamo però che, a questo livello, non è consentita la condivisione delle home degli utenti GNU/Linux sulle macchine Windows: di default essa è infatti riservata a utenti convalidati, e dunque solo se si adotta la direttiva `security = user`. Le risorse condivise andranno dunque specificate nella sezione `[share]`.

`security = user` il controllo viene eseguito a livello utente. Infatti per l'accesso vengono richiesti un nome utente e una password. Questi non vengono richiesti solo se l'utente Linux e l'utente Windows hanno lo stesso nome e la stessa password. Si tratta dell'impostazione di default di Samba. Ciò è consentito dal sistema di controllo detto SAM (acronimo di «System account manager»), basato su un archivio utenti che può essere gestito in tre modi:

a) mediante il file di testo ascii `/etc/samba/smbpasswd` che contiene le informazioni sugli utenti e le password criptate. Si tratta di una modalità poco versatile nel caso di reti Samba complesse. In ogni caso, in Samba3, dal momento che non è in grado di gestire alcune informazioni di Windows come i RID per i gruppi basati su NT, essa è stata sostituita dalla modalità TDBSM e viene mantenuto per retrocompatibilità.

In questo caso, la scelta del tipo di archivio utenti da utilizzare si effettua mediante la direttiva:

```
passdb backend = smbpasswd
```

Essa va precisata se la si vuole adottare, poiché di default Samba 3 adotta tdbsam. Inoltre, potrebbero doversi specificare le seguenti righe:

```
smb passwd file = percorso_del_file_delle_password
encrypt password
```

Il file delle password è di default /etc/samba/private/smbpasswd; la direttiva va inserita solo se questo file è collocato in una posizione non canonica. Inoltre, sempre di default le password vengono criptate, per compatibilità con le più recenti versioni di Microsoft Windows. La voce pertanto ha un senso solo se impostata a no, nel caso Samba debba dialogare con Windows 95 o versioni ancora precedenti; esistono comunque delle patch per i sistemi 9x, Me e 2k in /usr/doc/samba-*numero_versione*/docs/registry, costituite dai file di estensione .reg.

Si presuppone inoltre che l'utente Samba sia già stato aggiunto col comando smbpasswd. Per ovvi motivi di sicurezza, entrambi i file smbusers (presente in /etc/samba) e smbpasswd (presente in /etc/samba/private), devono essere accessibili sia in lettura che in scrittura per il solo utente root.

b) mediante archivi TDBSAM (Tiny database sam). Si tratta della modalità di default in Samba3, ideale per server locali che non necessitano di una replica del database interno. È adatta a domini con un numero limitato di utenti (non più di qualche centinaio, anche se il numero consigliato arriva a duecentocinquanta). Include nel file di database passdb.tdb tutte le informazioni precedentemente contenute in smbpasswd, e in aggiunta le informazioni SAM prima escluse.

In questo caso, la scelta del tipo di archivio utenti da utilizzare si può effettuare mediante la direttiva:

```
passdb backend = tdbsam,
```

che però non è necessario (e che anzi è deprecato) dichiarare poiché è attualmente il default. Vengono così creati i file /etc/samba/private/passdb.tdb, il database delle password che si gestisce con pdbedit, e /etc/samba/private/secrets.tdb, che contiene il SID (acronimo di «Security ID») di dominio, una sorta di identificativo del server. Poiché questo è il comportamento di default di Samba3, non serve decommentare la riga, già presente, poiché – come abbiamo appena ricordato – l'indicazione esplicita è deprecata. Non viene richiesta alcuna ulteriore configurazione.

Si presuppone inoltre che l'utente Samba sia già stato aggiunto col comando (che analizzeremo più in dettaglio sotto) pdbedit -a *nome_utente*.

c) mediante server locali o remoti NIS, LDAP, MySQL o altri ancora. Questa modalità è adatta a domini numericamente consistenti, che spesso richiedono la presenza di più PDC. In questi casi, poiché Samba non prevede la replica diretta degli archivi utente, una delle migliori opzioni è il ricorso a LDAP. Qui trascuriamo questa problematica.

In questo livello di sicurezza il nome dell'utente Samba deve obbligatoriamente coincidere con il nome di un utente definito nel sistema GNU/Linux, che, a sua volta, dev'essere sempre in grado di assegnare un proprietario valido ai file creati o copiati dall'utente nella risorsa condivisa. Se fosse impossibile soddisfare una tale condizione, si potrebbe ricorrere all'uso di un file di corrispondenza tra utenti GNU/Linux e utenti Samba, il cui nome viene indicato con la direttiva `username map = file_di_mappa` e che spesso prende il nome /etc/samba/smbusers. In questo file le indicazioni verranno fornite con la sintassi seguente:

```
nome_utente_linux = nome_utente_samba
```

`security = server` questo tipo di controllo risulta ormai obsoleto. Era usato quando Samba non era in grado di comportarsi come un domain member server, per cui l'utenza veniva controllata su

di un server esterno (solitamente un PDC MS-Windows, ma anche un PDC Samba), il cui nome NetBIOS veniva indicato con la direttiva: `password server = nome_server`. Inoltre esso è meno sicuro e causa un maggior carico al sistema rispetto al controllo esercitato mediante `security = domain`.

`security = domain` il controllo viene eseguito attraverso i controller del dominio. Il server Samba è concepito in un domain member server che, presentando un account della macchina («domain security trust account»), permette a tutte le richieste di autenticazione di passare per l'appunto attraverso i controller del dominio. Il tipo di controllo di accesso è dunque del tutto analogo a quello del caso di `security = user`, ma questa volta il server Samba va a inserirsi in un dominio MS-Windows NT/2000.

Tuttavia, i computer di una piccola rete domestica in genere non hanno un dominio proprio. In questa sede prescindiamo pertanto da questa problematica, che interessa piuttosto gli amministratori di sistemi di notevole ampiezza.

`security = ads` se si ha un ambiente Active Directory, la stazione Samba si unisce a un dominio come membro nativo dell'Active Directory, cioè come member server di una Active Directory di MS-Windows 200x. Sebbene le impostazioni di sicurezza limitino l'uso di protocolli di autenticazione compatibili con NT, il server Samba è in grado di unirsi a un ADS utilizzando l'autenticazione Kerberos5 e LDAP per l'archivio utenti: infatti Samba, in Active Directory member mode, è in grado di accettare i ticket di Kerberos. Prescindiamo qui anche da questa problematica, che interessa sempre gli amministratori di sistemi di notevole ampiezza.

Va infine ricordato che il livello di sicurezza per un server Samba singolo non può essere diversificato.

Gestione delle password

`smb passwd file` identifica il percorso del file che contiene le password, se si è scelto `security mode = user`. Di default è collocato in `/etc/samba/private`.

`encrypt password` cripta o meno le password. È un'opzione di sicurezza da scegliere qualora si adotti il backend `smbpasswd`, che va invece esclusa solo se si possiede Windows 95 o una versione ancora più vecchia. Se invece, per qualche ragione, si scegliesse di usare le password in chiaro anche con versioni successive di Windows, sarebbe necessario apportare una modifica al registro di sistema di Windows, ma ciò va a detrimento della sicurezza.

`null password` se impostato a `yes` permette di avere utenti Samba con parola d'ordine nulla (il valore predefinito è `no`).

`restrict anonymous` impedisce il collegamento anonimo. Per ragioni di sicurezza, è opportuno abilitare questa opzione.

`password level` di default, Samba esegue il matching delle password con tutti caratteri minuscoli, compresa la prima lettera. Questa voce permette di specificare, come nel caso di `username level`, quante lettere al massimo si possono rendere maiuscole per i tentativi di connettersi a una condivisione. Le si attribuisce un valore numerico, ad esempio: `password level = 2`.

Gestione del networking

`hosts allow` consente l'accesso solo agli IP inseriti.

`hosts deny` impedisce l'accesso agli IP inseriti, oppure a tutti se si usa la stringa `ALL`.

Come per le condivisioni NFS, queste due opzioni andrebbero gestite in coppia, in modo da negare a tutti l'accesso in `hosts deny` (con la stringa `ALL`) e consentirlo poi a quelli prescelti in `hosts allow`. Se vengono incluse in questa sezione, le voci `hosts allow` e `hosts deny` fanno riferimento a tutte le risorse, altrimenti hanno valore solo per la sezione nella quale sono inserite. Sono validi sia il nome `host` (ad esempio, `hosts allow = samiel`), sia l'indirizzo IP (ad esempio, `hosts allow = 192.168.0.2`), sia l'indicazione di una sottorete (ad esempio, `hosts allow = 192.168.1.*/255.255.0.0` o, in modo equivalente, `hosts allow = 200.200.`).

Inoltre, è opportuno che l'accesso a `localhost` sia concesso sempre, pena possibili malfunzionamenti della scansione delle risorse del server.

Si può usare anche l'operatore `EXCEPT` per indicare un'eccezione a una regola (ad esempio `hosts allow 192.168.1. EXCEPT 192.168.1.3`).

In assenza delle direttive `host allow` e `host deny`, l'accesso è concesso a tutti in modo predefinito.

Gestione degli accessi

`valid users` si imposta per consentire l'accesso esclusivamente agli utenti indicati.

`invalid users` si imposta per inibire definitivamente l'accesso agli utenti indicati.

A proposito delle opzioni `valid users` e `invalid users`, vale la pena di rilevare come Samba preveda due livelli di controllo di accesso alle risorse condivise: il primo, di carattere generale, si ottiene con l'impostazione del livello di sicurezza e in pratica stabilisce chi ha diritto di collegarsi al server; il secondo, di carattere più specifico, si effettua sulle singole risorse appunto mediante opzioni come `valid users`, `public write list` ecc., grazie alle quali si può stabilire quali utenti possono accedere a ogni singola risorsa.

`admin user` determina quali utenti, accedendo alla risorsa, possono eseguire operazioni da utente `root`.

`guest account` consente di fornire un account generico da considerare «guest» al quale può essere consentito l'accesso alle condivisioni (da usare nel caso di `security = share`). Il nome dell'utente viene scelto a piacere. Nelle sezioni corrispondenti alle risorse da condividere in tal modo andrà inserito il parametro `guest ok = yes`. Se qui non viene precisato nulla, verrà usato di default l'utente `nobody`.

`guest ok` impostato a `yes` rende disponibile la connessione anche a utenti anonimi.

`guest only` impostato a `yes` rende disponibile la connessione soltanto a utenti anonimi. Di default è settata a `no`.

`writable` opzione che permette di abilitare o disabilitare gli attributi di scrittura.

`read only` conferisce esclusivamente il permesso di lettura della risorsa.

Queste opzioni sono utilizzabili sia nella sezione `[global]` sia nella sezione `[share]`. I valori di default sono:

`read only = no`

`writable = yes`

`browseable` rende visualizzabili nell'elenco di Samba tutte le condivisioni disponibili. Se questo parametro, di default impostato a `yes`, viene impostato a `no`, ovviamente resta la possibilità di effettuare le condivisioni, ma è necessario conoscerne esattamente il nome.

`read list` permette di eseguire un override della configurazione base di una condivisione per la lista degli utenti specificati. In particolare, consente l'accesso con i soli diritti di lettura a una risorsa scrivibile.

`write list` permette di eseguire un override della configurazione base di una condivisione per la lista degli utenti specificati. In particolare, consente l'accesso con diritti di scrittura a una risorsa configurata per consentire accessi in sola lettura.

Entrambi questi override possono essere precisati non solo per utenti, ma anche per gruppi. I nomi dei gruppi sono preceduti dal simbolo di at (@) oppure dalla e commerciale (&). Ad esempio, si può avere:

```
write list = @webmasters @developers.
```

`max connections` specifica il numero massimo di connessioni a una condivisione. Settata di default a 0, permette connessioni illimitate alle risorse. A essa si attribuisce un valore numerico, ad esempio:
`max connections = 10.`

`dead time = nn` specifica il numero di minuti (*nn*) di inattività prima che il client venga disconnesso per evitare il consumo di risorse.

Gestione delle stampanti

`load printers` carica automaticamente (e perciò visualizza in Windows) tutte le stampanti presenti nel file `/etc/printcap`, che altrimenti dovranno essere specificate individualmente.

`printcap name` indica il percorso del file `printcap`, presente di default nella directory `/etc` e va inserito solo se il file si trova in una posizione diversa da quella canonica.

`printing` serve a specificare un sistema di stampa non standard e supporta `bsd`, `cups`, `sysv`, `plp`, `lprng`, `aix`, `hpux`, `qnx`. Poiché il sistema di default in Samba è CUPS ed esso lo è anche in Slackware, se esso è stato correttamente installato e configurato, qui non serve precisare nulla.

`show add printer wizard` impostato a `yes`, permette di far comparire l'icona per l'aggiunta di stampanti nella lista di condivisioni del server Samba quando si naviga in risorse di rete da un cliente MS-Windows.

Gestione del logging

`log file` indica il percorso del file di log. Di default crea un registro unico. In alternativa, è possibile impostare un file di log per ogni utente o uno per ogni client. Una direttiva come `log file = /var/log/samba/%m.log` imposta un file di log diverso per ogni client, che si chiamerà `nome_host.log`. mentre si avrà un diverso file di registrazioni per ogni utente usando opportunamente le variabili `%U` o `%u`.

`log level` indica il livello di dettaglio del file o dei file di log, da 0 (minimo) a 10 (massima verbosità). Più alto è il numero, maggiori saranno i dettagli e le risorse richieste. Se non si riscontrano problemi, va impostato fra 1 e 3; quest'ultima scelta garantisce un buon equilibrio fra lunghezza del file e informatività.

`max log size` indica la dimensione massima, in KB, del file di log, raggiunta la quale il file stesso viene rinominato con estensione `.old` e reinizializzato; il file `.old` eventualmente preesistente viene cancellato. Il valore predefinito di questo parametro è 5000 KB; il valore zero significa nessun limite di ampiezza (scelta non consigliabile per evitare una crescita abnorme del file o dei file delle registrazioni).

`syslog` richiede se utilizzare il `syslog`, presente in `/var/log/syslog`, per i log di Samba invece o parallelamente al logging diretto. Il valore specificato (da 0 a 10, di default 1) indica quali messaggi di log inviare a `syslog` e, in riferimento agli argomenti dell'opzione `log level`, che cosa loggare direttamente sui file di log interni. Per poter usare il `syslog` è necessario che Samba sia compilato (come in effetti è in Slackware) con l'opzione di configurazione `--with-syslog` e che `/etc/syslog.conf` sia configurato per gestire la facility daemon, con una riga tipo:

```
daemon.* /var/log/daemon.log
```

Ad esempio si può avere:

```
syslog = 20
```

```
log level = 30
```

Con queste opzioni i log di livello 0 e 1 (inferiori a 2) vengono mandati a `syslog`, quelli di livello 2 e 3 (compresi tra `syslog` e `log level`) vengono scritti sul file di log interno.

`syslog only` impostato a `yes`, fa sì che i log siano scritti esclusivamente su `syslog`.

`debug timestamp` fa in modo che in ogni riga di log generata siano inserite data e ora. Coincide con l'opzione `timestamp logs`. Il valore di default è `yes`. Per non specificare il timestamp scrivere: `debug timestamp = no`.

Gestione e ottimizzazione della connettività hardware

`interfaces` consente a Samba di usare diverse interfacce di rete. È utile solo nel caso che il server Samba risieda in più di una sottorete. Se nel computer sono presenti più interfacce di rete, in modo predefinito, Samba si mette in ascolto di richieste provenienti dagli indirizzi di rete corrispondenti alla rete della prima interfaccia che trova (di solito `eth0`). Per fare in modo che risponda alle richieste provenienti da più sottoreti si deve impostare esplicitamente questa opzione. Si può scegliere un indirizzo (ad esempio, `192.168.0.3`) o un intervallo di indirizzi IP (ad esempio, `192.168.0.3/17`) o una sottorete (ad esempio, `192.168.1.*` e allo stesso modo `192.168.1.*/255.255.255.0`, dove si usa una notazione con maschera di rete) oppure una determinata scheda di rete (ad esempio, `interfaces = eth0`).

`bind interfaces only` impostato a `yes`, limita l'accesso alle sottoreti corrispondenti alle interfacce raggiungibili secondo quanto specificato in `interfaces`. In tal caso si deve inserire tra le interfacce anche `127.0.0.1`, per consentire alle utilità di gestione degli utenti di potersi collegare al localhost e funzionare correttamente.

`#socket options#` stabilisce quali opzioni del socket vengono utilizzate nella comunicazione e può di conseguenza migliorare la risposta di Samba. Fra le opzioni, che vanno scelte in relazione alla struttura e alle caratteristiche della rete, troviamo:

`TCP_NODELAY`: fa sì che il server invii tanti pacchetti quanti servono per mantenere basso il ritardo. Questa opzione viene usata preferibilmente nelle connessioni telnet, ma migliora le prestazioni anche quando le richieste sono di piccola entità o quando Samba dialoga col sistema TCP/IP di Windows, che tende a rallentare le risposte.

`IPTOS_LOWDELAY`: questa opzione tende a diminuire il ritardo dovuto a un router e ad altri sistemi (non invece quello dovuto al server, per il quale vale l'opzione precedente).

`SO_RCVBUF`: aumenta i buffer di ricezione. I valori possibili sono multipli di 1024, ad esempio 4096 e 8192. In questo caso non ci sono regole fisse, ma solo una sperimentazione del funzionamento della rete.

`SO_SNDBUF`: aumenta i buffer di spedizione. I valori possibili sono multipli di 1024, ad esempio 4096 e 8192. Anche in questo caso non ci sono regole fisse, ma solo una sperimentazione del funzionamento della rete.

16.4.3 La sezione [homes]

La sezione [homes] è in realtà la prima sottosezione di [Share definitions]. Attivando questa sezione si condividono automaticamente tutte le directory home degli utenti, ogni qual volta il computer Windows si connette al computer GNU/Linux che le contiene. Questa sezione non dev'essere di conseguenza presente se si decide di condividere solo risorse scelte. La sezione [homes] viene utilizzata affinché ogni utente possa avere accesso a una propria directory personale sul server Samba che potrà anche coincidere con la directory personale GNU/Linux di quell'utente.

Ciò presenta tuttavia la controindicazione che ogni utente potrebbe visualizzare da Windows anche i «file punto», che in GNU/Linux rimangono nascosti (salvo impostazioni contrarie). Per ovviare a ciò, si può includere in questa sezione un'opzione di percorso: l'utente creerà una home separata dalla sua home di GNU/Linux, atta esclusivamente alla condivisione con Windows. Tale directory assumerà un nome a piacere, ad esempio /share, e andrà collocata preferenzialmente all'interno della home di GNU/Linux. Se si usano le impostazioni di default, questa home condivisa apparirà in Windows con il nome stesso dell'utente.

Le voci di questa sezione che è indispensabile conoscere sono le seguenti:

`comment` specifica un commento a piacere relativo alla risorsa condivisa, stringa che apparirà come una semplice descrizione in Windows.

`browseable` impostato a `yes`, fa sì che le risorse condivise del computer GNU/Linux appaiano in Windows. Per ragioni di sicurezza, appare più opportuno impostare a `no` questo parametro, di modo che la risorsa, sottratta alla scansione, non appaia con il nome /home a tutti gli utenti. In ogni caso, se condivise in maniera esplicita e selettiva, le home degli utenti appariranno comunque.

`writable` definisce l'accesso in scrittura per la risorsa condivisa.

`write list` se si sceglie l'opzione `writable = yes`, è possibile inserire in questa voce gli utenti o i gruppi che potranno, in maniera esclusiva, scrivere sulla risorsa. I nomi dei gruppi vanno fatti precedere dal simbolo di at (@).

`read only` definisce l'accesso in sola lettura

`create mode` definisce i permessi di default per la creazione di un file (il valore preimpostato è 0750).

`path` l'indicazione di un path fa sì che questa directory non coincida con quella GNU/Linux dell'utente (come da impostazione predefinita).

16.4.4 La sezione [printers]

Qui si impostano, tutti in una sola volta, i parametri di configurazione di tutte le stampanti definite nel sistema GNU/Linux, a condizione che nella sezione [global] siano state inserite le direttive seguenti:

```
load printers = yes
printcap name = /etc/printcap
```

In alternativa, si potrebbero definire le varie stampanti come singole risorse condivise. Dal punto di vista della struttura, questa sezione funziona in modo analogo a quella sulla condivisione delle home. L'impostazione classica è la seguente:

```
[printers]
comment = stampanti
path = /var/spool/samba
browseable = no
printable = yes
public = yes
writable = no
```

Qui occorre notare che l'eventuale parametro `path` serve a impostare una directory per la coda di stampa diversa da `/tmp`, che è quella predefinita.

Altra direttiva da segnalare è `printable` con la quale si attiva la coda di stampa.

Se si adotta la direttiva `browseable = yes`, è necessario che anche CUPS sia configurato per consentire il browsing delle stampanti. A tal fine bisogna accertarsi fundamentalmente che nella sezione `Browsing Options` del file `/etc/cups/cupsd.conf` la riga

```
Browsing On
```

sia decommentata. In aggiunta, si possono definire gli IP autorizzati o interdetti al browsing ricorrendo, più sotto in questa medesima sezione, alle direttive `BrowseAllow` e `BrowseDeny`.

16.4.5 La sezione [tmp]

La sezione `[tmp]` specifica la directory dei file temporanei, dedicata cioè allo swap del programma. Tra le sue voci citiamo:

`comment` specifica un commento a piacere relativo alla risorsa condivisa.

`path` definisce il percorso assoluto della cartella dei file temporanei, in genere `/tmp`.

`public` se impostata a `yes`, rende la risorsa pubblicamente accessibile, cioè a tutti gli utenti.

`read only` definisce l'eventuale accesso in sola lettura.

16.4.6 La sezione [public]

La sezione `[public]` specifica delle risorse condivise a cui gli utenti possono accedere senza specificare alcuna password. Tra le sue voci citiamo:

`comment` specifica un commento a piacere relativo alla risorsa condivisa.

`path` definisce il percorso assoluto della risorsa condivisa.

`public` se impostata a `yes`, rende la risorsa pubblicamente accessibile, cioè a tutti gli utenti.

`guest ok` se impostata a `yes`, ammette qualsiasi utente, col nome prescelto nella sezione `[global]` o, altrimenti, consente l'accesso all'utente `nobody`.

`writable` se impostata a `yes`, definisce l'accesso in scrittura per la risorsa condivisa

`write list` rende la risorsa scrivibile solo da utenti segnalati o dai membri di un gruppo definito, segnalati in questa opzione.

16.5 Alcuni esempi di /etc/samba/smb.conf

Presentiamo qui alcuni semplici esempi di configurazione di /etc/samba/smb.conf, tutti relativi a server di tipo stand-alone (abbiamo già precisato che in questa sede non verrà considerata la configurazione per server del membro del dominio). Un server di tipo stand-alone può essere sia un server di tipo workgroup sia un membro di un ambiente workgroup. Un server stand-alone non rappresenta un domain controller e non partecipa a un dominio in alcun modo.

16.5.1 Configurazione di base

Questa può ritenersi una configurazione standard per una condivisione senza limiti e senza controlli di una serie di risorse condivise (qui ricorriamo esemplificativamente a due directory che chiamano share e public). La direttiva security = share consente l'accesso senza password agli utenti. Modificandola in security = user verrà chiesta la password agli utenti, a meno che l'utente GNU/Linux e l'utente Windows non abbiano lo stesso nome e la stessa password. Anche il ricorso alla stampante è incondizionato.

```
#===== Global Settings=====
[global]
    workgroup = workgroup
    netbios name = darkstar
    security = share
    load printers = yes
    log file = /var/log/samba.%m
    max log size = 50

#===== Share Definitions=====
[share]

[condivisioni]
    comment = Condivisioni
    path = /share
    browseable = no
    writable = yes

[pubblico]
comment = Condivisioni
    path = /documenti/public
    browseable = no
    writable = yes

[printers]
    comment = All Printers
    path = /var/spool/samba
    browseable = no
    guest ok = no
    writable = no
    printable = yes
```


16.5.2 Condivisione delle home degli utenti

La condivisione delle home degli utenti è possibile solo in base a un accesso controllato, per utenti convalidati. È perciò necessario impostare la direttiva `security = user`, che è attualmente il default di Samba.

```
##### Global Settings#####
[global]
    workgroup = workgroup
    netbios name = darkstar
    security = share
    load printers = yes
    log file = /var/log/samba.%m
    max log size = 50

##### Share Definitions#####
[share]

[homes]
    comment = Home Directories
    browseable = yes
    read only = no

[printers]
    comment = All Printers
    path = /var/spool/samba
    browseable = no
    guest ok = no
    writable = no
    printable = yes
```

In questo caso, le home degli utenti, in base alla direttiva `read only = no`, possono anche essere scritte da Windows.

16.5.3 Condivisione pubblica in sola lettura

Una directory accessibile a tutti gli utenti si chiama «pubblica», tale per cui tutti gli utenti possono creare, modificare, cancellare qualsiasi file. In questo esempio il file `/etc/samba/smb.conf` è configurato per implementare una condivisione pubblica in sola lettura della directory `/export`. Il parametro `security = share` rende la condivisione anonima, il parametro `read only = yes` la rende in sola lettura.

```
[global]
    workgroup = Workgroup
    [netbios name = darkstar]
    security = share

[data]
```

```
comment = Samba Server
path = /export
read only = yes
guest only = yes
```

16.5.4 Condivisione pubblica in lettura e scrittura

In questo esempio il file `/etc/samba/smb.conf` è configurato per implementare una condivisione pubblica in lettura e scrittura della directory `/export`. Il parametro `read only = no` la rende in lettura e scrittura. Le ulteriori opzioni `force user = nome_utente` e `force group = nome_gruppo` vengono aggiunte per rinforzare la proprietà di qualsiasi nuovo file ammesso alla condivisione (senza cioè tener conto dell'utente specifico, a cui viene assegnata la coppia utente e gruppo `anybody:users`). Si tratta verosimilmente del tipo di condivisione meno sicura.

```
[global]
workgroup = Workgroup
[netbios name = darkstar]
security = share

[data]
comment = Samba Server
path = /export
force user = anybody
force group = users
read only = no
guest ok = yes
```

16.5.5 Una directory accessibile a tutti gli utenti in lettura e scrittura

In questo esempio il file `/etc/samba/smb.conf` è configurato per implementare una condivisione anonima in lettura e scrittura della directory `/export`. Qui, la direttiva `guest = ok` rende possibile l'accesso anonimo. Eventuali file creati o copiati nella directory saranno di proprietà dell'utente indicato in `guest account` in caso di accesso anonimo, oppure dell'utente effettivo in caso esso sia registrato in GNU/Linux.

```
[public]
comment = directory pubblica
browseable = yes
guest ok = yes
path = /usr/local/public
writable = yes
```

16.5.6 Una directory accessibile indiscriminatamente a tutti gli utenti in lettura e scrittura

In questo esempio il file /etc/samba/smb.conf è configurato per implementare una condivisione anonima in lettura e scrittura sulla directory /export. C'è però una differenza rispetto all'esempio precedente, dovuto all'aggiunta della direttiva `guest only = ok`. In questo caso infatti il proprietario è sempre l'utente fittizio ospite, anche nel caso che l'utente collegato sia già registrato come utente di GNU/Linux; in questo caso quindi tutti gli utenti possono fare tutte le operazioni con qualsiasi file presente nella directory condivisa. Un modo alternativo per ottenere lo stesso risultato è quello di usare la direttiva `force user = nome_utente`; in questo modo Samba assegna lo stesso nominativo di utente a chiunque si connetta alla risorsa.

```
[temp]
comment = directory pubblica plus
browseable = yes
guest ok = yes
guest only = yes
path = /usr/local/public
writable = yes
```

16.5.7 Condivisione sicura in lettura e scrittura

In questo esempio il file /etc/samba/smb.conf è configurato per implementare una condivisione sicura in lettura e scrittura. Impostando la direttiva `security = user`, si forza Samba all'autenticazione dei collegamenti client. Qui la condivisione [homes] non possiede una direttiva `force user` o `force group`, al contrario della condivisione [public] vista in uno degli esempi precedenti, ma utilizza le informazioni dell'utente autenticato per qualsiasi file creato.

```
[global]
workgroup = DOCS
netbios name = DOCS_SRV
security = user
printcap name = cups
disable spools = yes
show add printer wizard = no
printing = cups

[homes]
comment = Home Directories
valid users = %S
read only = No
browseable = No

[public]
comment = Data
path = /export
```

```
force user = samiel
force group = users
guest ok = yes
```

16.5.8 Una directory accessibile solo ad alcuni utenti

In questo esempio la sezione [private] del file /etc/samba/smb.conf è configurata per implementare una condivisione sicura in lettura e scrittura di una directory (/home/samiel/riservato) accessibile soltanto ad alcuni utenti, precisati con la direttiva valid users.

```
[private]
comment = documenti riservati
browseable = yes
public = no
path = /home/samiel/riservato
writable = yes
valid users = utente1 utente2
```

La direttiva comment serve, al solito, ad associare una descrizione alla risorsa condivisa. L'opzione browseable (qui impostata a yes) permette di rendere visibile la risorsa agli utenti che si connettono al server. L'opzione public è un sinonimo di guest ok; in questo caso non si vuole che la risorsa sia accessibile per l'utente generico. L'opzione path permette di indicare il percorso della risorsa sul sistema GNU/Linux. L'opzione writable (qui impostata a yes) serve a concedere l'accesso in scrittura. La direttiva valid users indica quali sono gli utenti che possono accedere alla risorsa. È anche possibile indicare gruppi di utenti GNU/Linux con la sintassi @nome_gruppo e gruppi di utenti NIS (se in rete è presente un server NIS) con la sintassi &nome_gruppo, e infine entrambi, con la sintassi +&nome_gruppo o &+nome_gruppo.

16.5.9 Una directory privata con permessi preimpostati

In questo esempio la sezione [private] del file /etc/samba/smb.conf è configurata per implementare una condivisione in lettura e scrittura di una directory (/usr/local/private) accessibile in scrittura soltanto ad alcuni utenti.

```
[private]
comment = directory privata
browseable = yes
path = /usr/local/private
writable = no
public = no
write list = pippo pluto
create mask = 0644
directory mask = 0644
follow symlinks = no
```

In questo esempio abbiamo una directory accessibile in sola lettura, dove tuttavia la presenza della direttiva `write list` permette di impostare permessi di scrittura per gli utenti indicati, indipendentemente da quanto specificato negli altri parametri. Si deve però aggiungere che i permessi impostati a livello di sistema su una risorsa hanno sempre il sopravvento su quanto specificato nel file di configurazione di Samba; nel caso considerato, se `/usr/local/private` fosse di proprietà dell'utente `root` e i permessi impostati a `600`, gli altri utenti non potrebbero comunque né leggere né scrivere alcunché in quella directory condivisa. Se si volesse innalzare un utente al rango di `root`, lo si potrebbe fare con la direttiva `admin users = nome_utente`; in questo modo egli non risentirebbe di eventuali limitazioni dovute ai permessi sui file. Le ultime due direttive servono a indicare i permessi con cui verranno creati file e directory all'interno della risorsa condivisa; il valore predefinito è `0755`. L'ultima direttiva presente nell'esempio, `follow symlinks`, impostata a `no` impedisce che vengano seguiti i collegamenti simbolici, evitando così che chi accede alla condivisione possa accedere anche a file che si trovano al suo esterno.

16.5.10 Condivisione di un CD-Rom

In questo esempio, si consente agli utenti di montare e smontare il CD-Rom. Le opzioni `preexec` e `postexec` permettono di impostare un'azione che verrà eseguita rispettivamente alla connessione e alla disconnessione dalla risorsa condivisa. Nel caso specifico, allorché verrà montata la risorsa, verrà montato automaticamente anche il CD-Rom, e quando la risorsa verrà smontata, verrà smontato anch'esso.

```
[cd]
comment = CD-ROM
preexec = mount /mnt/cdrom
postexec = umount /mnt/cdrom
browseable = yes
public = yes
path = /mnt/cdrom
writable = no
```

Se si vuole evitare che possano connettersi utenti sprovvisti dei privilegi per montare e smontare il CD, le direttive `preexec` e `postexec` possono essere sostituite rispettivamente con `root preexec` e `root postexec`, che svolgono lo stesso compito, ma con i privilegi dell'utente `root`.

16.6 Le utilità di Samba

Samba, come abbiamo visto, mette a disposizione diversi strumenti per svolgere diverse funzioni di controllo. Vediamo i principali con le opzioni più comuni, distinguendo quelli che operano dal lato del server e quelli che operano dal lato del client.

a) Utilità lato server:

smbstatus visualizza lo stato corrente del server Samba

smbtree visualizza un albero con tutti le risorse condivise

findsmb fornisce informazioni sui server Samba presenti in rete

smbpasswd gestisce utenti e password

testparm consente di controllare la correttezza sintattica di `/etc/samba/smb.conf`

smbcontrol consente di interagire con i demoni di Samba

nmblookup verifica il nome NetBios di un computer Windows e ne costituisce l'IP

pdbedit gestisce utenti e password

net amministra i server remoti Samba e CIFS. Trascuriamo qui questa utilità

SWAT configura via browser il file `/etc/samba/smb.conf`

b) Utilità lato client:

smbclient offre un accesso di tipo FTP dai client Linux ai servizi Samba (e CIFS) sui server

smbmount consente di montare un file system smbfs

smbumount consente di smontare un file system smbfs

16.6.1 L'utilità `smbstatus`

Il comando `smbstatus` visualizza un report sullo stato delle connessioni del server Samba e indica chi lo sta utilizzando; è utile pertanto per debugging e gestione del servizio. Senza specifiche opzioni, `smbstatus`, oltre alla versione di Samba, visualizza semplicemente lo stato delle connessioni, elencando il tipo di servizio, il numero dei processi attivi (PID), l'IP delle macchine connesse e data e ora della connessione.

```
root@darkstar:~# smbstatus

Samba version 3.0.22
PID      Username      Group          Machine
-----
Service  pid          machine       Connected at
-----
IPC$     8201         zetaz1        Wed Jul 19 16:57:59 2006
IPC$     5419         192.168.0.4   Wed Jul 19 15:11:40 2006
IPC$     7015         pc-zsq2z7ufh2z0 Wed Jul 19 16:29:44 2006

No locked files
```

Il comando può essere integrato con diverse opzioni:

`-b`: attiva una visualizzazione sintetica

`-d`: attiva una visualizzazione estesa

`-p`: elenca i processi correnti di Samba

`-s path_di_smb.conf`: specifica una posizione diversa da quella canonica per il file di configurazione

`-u nome_utente`: mostra lo stato della connessione per un utente specifico

Con l'opzione `-p`, il comando visualizza i processi attivi del demone `smbd`, indicandone il numero (PID):

```

root@darkstar:~# smbstatus -p
Samba version 2.2.7
Service      uid      gid      pid      machine
-----
13388
13039
[...]

```

16.6.2 L'utilità smbtree

Il comando `smbtree` fornisce in forma di albero l'elenco delle risorse condivise:

```

root@darkstar:~# smbtree
Password:
WORKGROUP
  \\DARKSTAR                               Samba Server
    \\DARKSTAR\samiel                       Home Directories
    \\DARKSTAR\ADMIN$                       IPC Service (Samba Server)
    \\DARKSTAR\IPC$                         IPC Service (Samba Server)
  \\BRIGHTSTAR                             Brightstar
    \\BRIGHTSTAR\C$                         Condivisione predefinita
    \\BRIGHTSTAR\ADMIN$                     Amministrazione remota
    \\BRIGHTSTAR\D$                         Condivisione predefinita
    \\BRIGHTSTAR\IPC$                       IPC remoto

```

16.6.3 L'utilità find smb

Il comando `findsmb` (in realtà uno script in Perl) elenca le macchine che rispondono all'interrogazione di Samba in una sottorete specifica, fornendone l'indirizzo IP, il nome NetBIOS, il workgroup o nome del dominio, il sistema operativo e la versione. Se non si specifica alcuna sottorete, allora viene usata quella locale.

IP ADDR	NETBIOS NAME	WORKGROUP/OS/VERSION
192.168.35.10	MINESET-TEST1	[DMVENGR]
192.168.35.55	LINUXBOX	*[MYGROUP] [Unix] [Samba 3.0.2]
192.168.35.56	HERBNT2	[HERB-NT]
192.168.35.63	GANDALF	[MVENGR] [Unix] [Samba 2.0.5a for IRIX]
192.168.35.65	SAUNA	[WORKGROUP] [Unix] [Samba 1.9.18p10]
192.168.35.71	FROGSTAR	[ENGR] [Unix] [Samba 2.0.0 for IRIX]
192.168.35.78	HERBDHCP1	+ [HERB]
192.168.35.88	SCNT2	+ [MVENGR] [Windows NT 4.0] [NT LAN Manager 4.0]
192.168.35.93	FROGSTAR-PC	[MVENGR] [Windows 5.0] [Windows 2000 LAN Manager]
192.168.35.97	HERBNT1	* [HERB-NT] [Windows NT 4.0] [NT LAN Manager 4.0]

Il simbolo + davanti al nome del gruppo di lavoro indica che quella macchina è local master browser per tale gruppo; il simbolo * davanti al nome del gruppo di lavoro indica che quella macchina è domain master browser per tale gruppo.

16.6.4 L'utilità smbpasswd

Se usato da root, il comando `smbpasswd` gestisce le password degli utenti Samba col vecchio backend, utilizzando una struttura del tipo plain ASCII text che include MS Windows LanMan e NT account e le informazioni password codificate. Un nuovo utente, che deve comunque essere un utente GNU/Linux già presente nel sistema (più esattamente, tale utente dev'essere presente in `/etc/passwd`), si crea col comando: `smbpasswd -a nome_utente`. Ne viene richiesta due volte, per sicurezza, la password, e, se il processo si conclude felicemente, leggeremo l'avviso «Added user *nome_utente*». Per cancellare un utente si usa il comando: `smbpasswd -x nome_utente` (se non viene indicato alcune nome utente, si fa riferimento in modo predefinito all'utente GNU/Linux che esegue il comando).

Poiché Samba non fa ricorso al sistema shadow delle password, le sue password sono del tutto indipendenti da quelle del sistema GNU/Linux. Anzi, per ovvi motivi di sicurezza, è opportuno – soprattutto per l'utente root – che le due password siano diverse.

Un utente può anche essere disabilitato e nuovamente abilitato col medesimo comando ma, rispettivamente, con le opzioni `-d` ed `-e`. Per modificare una password già impostata si usa il comando: `smbpasswd [-r server_samba] -U nome_utente`. Viene chiesta la vecchia password e, per due volte, quella nuova. L'opzione `[-r server_samba]` è opzionale, in quanto serve a precisare un determinato server per cui vale quella password; se viene omessa, il valore di default è localhost.

Usato dagli utenti non root, consente loro di modificare la password, sempre se è stato adottato il vecchio backend.

16.6.5 L'utilità testparm

Il comando `testparm` è uno strumento atto a verificare che la sintassi di `/etc/samba/smb.conf` sia corretta. Naturalmente, ciò non garantisce ancora il buon funzionamento della connessione, perché si limita a un'analisi formale del file di configurazione. Con l'opzione `-v` `testparm` analizza anche le opzioni non usate. Se si precisa lo hostname e l'indirizzo IP di un host presente nella sezione `hosts allow` o `hosts deny` di `/etc/samba/smb.conf`, `testparm` verifica se questo IP è autorizzato ad accedere al server Samba.

16.6.6 L'utilità smbcontrol

Il comando `smbcontrol` consente di interagire con i demoni di Samba `smbd`, `nmbd` e `winbindd`. La sua sintassi è: `smbcontrol [destinazione] [tipo di messaggio] [parametro]`.

Le opzioni `-i` e `-s` servono per determinare la modalità di effettuazione. In dettaglio, l'opzione `-s` `[file di configurazione]` specifica un file contenente dei dettagli di configurazione richiesti dal server, mentre l'opzione `-i` avvia una procedura interattiva, da precisare nello standard input. Una riga vuota o il comando `q` determinano l'uscita dal programma.

La destinazione si precisa indicando il demone (`smbd`, `nmbd` o `winbindd`), oppure il numero di un processo (PID). La destinazione `smbd` invia il messaggio a tutti i demoni `smbd`; la destinazione `nmbd` invia il messaggio al demone `nmbd` specificato nel file `thenmbd.pid`. La destinazione PID invia il messaggio soltanto a quel processo.

Fra i numerosi parametri ricordiamo:

`close-share`: ordina a `smbd` di chiudere le condivisioni dei client per una determinata condivisione. Non vengono affette le altre condivisioni. L'uso della wildcard `*` fa sì che vengano chiuse tutte le condivisioni aperte.

ping: invia un determinato numero di ping e attende lo stesso numero di messaggi replica. Può essere inviato a qualsiasi destinazione.

shutdown: spegne il demone specificato, che può essere sia `smbd` sia `nmbd`.

pool-usage: stampa un resoconto sull'uso della memoria da parte del processo specificato. È disponibile sia per `smbd` sia per `nmbd`.

reload-config: obbliga il demone a rileggere il file di configurazione (di default `/etc/samba/smb.conf`). Può essere inviato a `smbd`, `nmbd` o `winbindd`.

16.6.7 L'utilità nmblookup

Il comando `nmblookup` permette di trovare il numero IP di una macchina fornendo il nome NetBIOS usando NetBIOS su interrogazioni TCP/IP, interrogazioni che vengono effettuate su UDP. Alcune sue opzioni consentono di dirigere le interrogazioni verso una particolare area di indirizzi IP o a un computer specifico. Il comando si usa con la sintassi: `nmblookup [opzioni] [nome_NetBIOS]`. Si ottiene, ad esempio:

```
root@darkstar:~# nmblookup brightstar
querying brightstar on 192.168.0.255
192.168.0.4 brightstar<00>
```

16.6.8 L'utilità pdbedit

Il programma `pdbedit` è in grado di gestire gli account che si trovano all'interno del database SAM, il database degli utenti Samba. Tutti i backend sono supportati, inclusi `smbpasswd`, LDAP, NIS+ e la libreria del database `tdb`. La sua sintassi è: `pdbedit [opzione]`.

Innumerevoli sono le sue funzioni. Tra le principali ricordiamo:

aggiunge un utente: `pdbedit -a -u nome_utente` (richiede due volte la password dell'utente)

cancella un utente: `pdbedit -v -L nome_utente`

elenca gli utenti: `pdbedit -L`

elenca gli utenti in formato esteso (= verbose): `pdbedit -L -v`

Fra le numerose opzioni (per visionarle tutte, ricorrere alla pagina `man`), citiamo:

-u *nome_utente* specifica il nome dell'utente che deve compiere le operazioni richieste. È necessario per i comandi di aggiunta, rimozione e modifica.

-f *nome_completo* gestisce il nome completo dell'utente (scelta peraltro del tutto opzionale), da porre fra apici doppi (ad esempio, -f "Simo Sorce")

-h *home* specifica la home dell'utente, indicandone il path completo sempre fra apici doppi (ad esempio, -h "\\\\BERSERKER\\sorce")

-D *drive* specifica, sempre fra apici doppi, la lettera dell'unità di Windows che dev'essere usata per mappare la home dell'utente (ad esempio, -d "H:")

-m *macchina* usata in congiunzione con l'opzione -a, aggiunge un account fiduciario per una macchina invece che per un utente. Con la specificazione ulteriore dell'opzione -u fornisce il nome della macchina (ad esempio, `pdbedit -a -m -u w2k-wks`)

-x cancella un account, relativo a un utente o a una macchina, dal database. Il nome dell'utente va specificato con l'opzione -u (ad esempio, `pdbedit -x -u bob`)

- i *passwd-backend* usa un backend diverso per richiamare gli utenti rispetto a quello specificato in `/etc/samba/smb.conf`. Può essere usato per importare i dati relativi agli utenti nel database locale, facilitando la migrazione da un backend per le password a un altro (ad esempio, `pdbedit -i smbpasswd:/etc/samba/private/smbpasswd.old`).
- e *passwd-backend* esporta tutti gli utenti attualmente disponibili nel backend delle password specificato (ad esempio, `pdbedit -e smbpasswd:/root/samba-users.bac`).
- g fa sì che le opzioni -i (in-backend) ed -e (out-backend) si applichino alla mappatura di un gruppo piuttosto che al database degli utenti.
- b *passwd-backend* usa un backend differente da quello adottato di default per le password (ad esempio, `pdbedit -b xml:/root/pdb-backup.xml -l`)

16.6.9 L'utilità smbclient

Il comando `smbclient` consente, innanzitutto, di elencare le condivisioni disponibili su un determinato host se accompagnato dall'opzione `-L`. La sintassi del comando è la seguente: `smbclient -L nome_host`, dove il *nome_host* è quello conferito alla voce `netbios_name` di `/etc/samba/smb.conf` (se esso è stato definito in `/etc/hosts`) oppure un indirizzo IP. L'output potrebbe essere del tipo:

```
bash-3.00# smbclient -L darkstar
Password:
Domain=[DARKSTAR] OS=[Unix] Server=[Samba 3.0.14a]

      Sharename      Type      Comment
      -----      -
IPC$                IPC       IPC Service (Samba Server)
ADMIN$              IPC       IPC Service (Samba Server)
samiel              Disk      Home Directories
Domain=[DARKSTAR] OS=[Unix] Server=[Samba 3.0.14a]

      Server          Comment
      -----
Workgroup            Master
WORKGROUP           BRIGHTSTAR
```

Tra le informazioni che emergono, notiamo che `Sharename` indica il nome che identifica la risorsa condivisa, `Type` fornisce delle indicazioni sul tipo della risorsa (disco oppure stampante), `Comment` aggiunge ulteriori indicazioni.

Le colonne `Server` e `Comment` forniscono i nomi degli altri host appartenenti alla rete con delle informazioni aggiuntive, e infine le colonne `Workgroup` e `Master` indicano il nome del gruppo e di altri eventuali gruppi e il domain master del gruppo.

Il comando `smbclient` consente invece di connettersi a una determinata risorsa se viene specificato il nome di un servizio, secondo la sintassi: `smbclient [//server/servizio]`, ad esempio: `smbclient //server_samba/public`. Se la connessione ha successo, si ha a disposizione un'interfaccia testuale (un prompt) del tutto simile a quella del programma `ftp`, dove è possibile eseguire una serie di comandi

analoghi (get, put, cd, pwd, ecc.) per prelevare file dal server in direzione della macchina locale, per prelevare file dalla macchina locale in direzione del server, per ricevere informazioni dal server e così via. Avremo così un prompt del tipo:

```
root@darkstar:~# smbclient //brightstar/C
Password:
Domain=[BRIGHTSTAR] OS=[Windows 5.1] Server=[Windows 2000 LAN Manager]
smb: \>
```

L'opzione `-I` seguita da un numero IP serve a indicare l'indirizzo del server a cui ci si vuole connettere.

L'opzione `-U` seguita dalla stringa `username%password` automatizza l'accesso. Se la password non viene specificata, verrà comunque richiesta. In alternativa, è possibile ricorrere a un file di credenziali (di solo testo) che contiene in chiaro il nome dell'utente e la sua password. Ad esempio, si può creare il file `/etc/samba/smbcred.brightstar` (l'estensione è il nome del computer Windows o della condivisione) avente per contenuto le seguenti righe:

```
username = nome_utente
password = password
```

Per sicurezza, l'amministratore può restringere l'accesso a questo file per prevenirne la lettura da parte di utenti non autorizzati, coi seguenti comandi:

```
chmod 600 /etc/samba/smbcred.brightstar
chown root.root /etc/samba/smbcred.brightstar
```

16.6.10 L'utilità `smbmount`

Il comando `smbmount` consente il montaggio delle risorse condivise come se fossero risorse locali. La sintassi del comando è: `smbmount [risorsa] [punto_di_montaggio] [-o opzioni]`. Avremo perciò, ad esempio:

```
smbmount //servizio_samba/directory /mnt/directory_samba \
-o username=nome_utente%password
```

o ancora:

```
smbmount //192.168.0.2/homes /punto_di_montaggio -o \
username=nome_utente,dmask=777,fmask=777
```

16.6.11 L'utilità `smbumount`

Il comando `smbumount` consente lo smontaggio delle risorse condivise. La sua forma è semplicemente: `smbumount punto_di_montaggio`.

16.6.12 Altre tecniche di montaggio e smontaggio

I due comandi `smbmount` e `smbumount` possono anche essere sostituiti dai tradizionali comandi di montaggio (`mount`) e smontaggio (`umount`) di unità sulla directory di montaggio. Avremo così, ad esempio, il comando:

```
mount -t smbfs //192.168.0.2/homes /punto_di_montaggio -o \
username=nome_utente,dmask=777,fmask=777
```

Qui il nome dell'host può essere sostituito da un indirizzo IP se è attivo il servizio di risoluzione dei nomi o se l'host è elencato in `/etc/hosts`.

Per smontare la condivisione si usa, al solito, il comando: `umount /punto_di_montaggio`.

Si può rendere stabile il montaggio inserendo in `/etc/fstab` una riga del tipo:

```
//serv_samba/public /mnt/samba smbfs username=nome_utente%password
```

posto che la directory di montaggio sia `/mnt/samba`, oppure:

```
//192.168.0.3/C /mnt/samba smbfs defaults, \
credentials=/etc/samba/smbcred.brightstar, gid=102,fmask=660,dmask=770 0 0+
```

dove si vede anche il ricorso a un file di credenziali.

In questo caso, il montaggio da console si effettua col comando: `mount /mnt/samba`.

Infine, si può attivare il servizio di automount inserendo in `/etc/auto.misc` una riga del tipo:

```
samba -fstype=smbfs,credentials=/etc/samba/smbcred.brightstar, \
gid=102,fmask=660,dmask=770 ://192.168.0.3/C
```

Alcuni browser supportano il protocollo Samba: fra gli altri Konqueror, nel quale si possono visualizzare le risorse condivise all'indirizzo: `smb:/`.

16.7 La stampa con Samba

Se il nostro obiettivo è stampare da GNU/Linux con una stampante installata su di una macchina Windows o, al contrario, stampare da Windows con una stampante installata su di una macchina GNU/Linux, le operazioni da compiere sono davvero poche e semplici.

La configurazione diventa leggermente più complessa (ma qui accenneremo solo a questa problematica) se si fanno intervenire considerazioni sulla sicurezza e sulla privacy nella stampa. In questo caso, si potrebbe fra l'altro impostare Samba in modo che gli utenti fossero in possesso di uno print spooler proprio, non posizionato in un percorso pubblico. Se si verificasse un errore in un particolare processo, gli altri utenti non avrebbero così la possibilità di accedere al file.

Samba supporta vari sistemi di stampa (`bsd`, `cups`, `sysv`, `plp`, `lprng`, `aix`, `hpux`, `qnx`); i più consolidati sono l'ormai obsoleto BSD (di cui non ci occupiamo in questa sede) e CUPS, attualmente il default in Slackware. Si presuppone naturalmente che CUPS sia installato e che la stampante sia stata felicemente configurata nel sistema.

16.7.1 La configurazione di `/etc/samba/smb.conf` per la stampa

Il seguente esempio mostra una configurazione di base di `/etc/samba/smb.conf` per il supporto a CUPS:

```
[global]
    load printers = yes
;   printcap name = /etc/printcap
;   printing = cups

#===== Share Definitions=====

[printers]
    comment = All Printers
    path = /var/spool/samba
    browseable = no
    guest ok = no
    writable = no
    printable = yes
```

Qui, nella sezione [global], abbiamo una prima riga che carica automaticamente tutte le stampanti configurate con CUPS e presenti in /etc/printcap senza che ci sia il bisogno di specificarle individualmente. La direttiva va impostata a no solo in server dove gli amministratori intendono effettuare un controllo preciso di ciò che visibile agli utenti. Se il file /etc/printcap fosse collocato in una posizione non canonica, allora andrebbe precisata la seconda riga dell'esempio, attualmente commentata. Se non si intende adottare CUPS come sistema di stampa, allora andrebbe precisata la terza riga dell'esempio, attualmente commentata

Nella sezione [printers], dopo una riga meramente descrittiva, comment, vengono indicati lo spooling di Samba, comune per tutte le operazioni e tutti gli utenti, e, con la direttiva browseable = no, il fatto che la risorsa non verrà visualizzata in Windows. Questa configurazione è resa abbastanza sicura dalla direttiva guest ok = no, che impedisce utilizzi anonimi della stampante; la direttiva opposta può essere espressa sia nella forma di guest ok = yes, sia nella forma di set public = yes. Infine, la risorsa è posta ovviamente come non scrivibile.

16.7.2 Alcuni esempi di configurazione

Server di stampa accessibile a tutti gli utenti

Nell'esempio che segue il file /etc/samba/smb.conf è configurato per implementare un server di stampa accessibile indiscriminatamente a tutti gli utenti, come si evince dalla presenza della direttiva guest = ok.

```
[printers]
    comment = All Printers
    path = /var/spool/samba
    guest ok = yes
    printable = yes
    browseable = no
```

Server di stampa anonimo

Nell'esempio che segue il file `/etc/samba/smb.conf` è configurato per implementare un server di stampa anonimo. Impostando `browseable = no`, la stampante non viene elencata nelle risorse di rete in Windows. Anche se nascosta dal browsing, la stampante può essere comunque configurata: collegandosi al server, il client può accedere alla stampante se lo stesso client fa parte del workgroup indicato. In questo esempio si presume anche che il client abbia installato il corretto driver della stampante locale, in quanto la direttiva `use client driver` è impostata a `yes`: il server Samba non ha allora alcuna responsabilità per la condivisione dei driver di stampa col client.

```
[global]
workgroup = WORKGROUP
netbios name = darkstar
security = share
disable spools= yes
show add printer wizard = no
printing = cups

[printers]
comment = All Printers
path = /var/spool/samba
guest ok = yes
printable = yes
use client driver = yes
browseable = yes
```

Server di stampa ad amministrazione controllata

Nell'esempio che segue il file `/etc/samba/smb.conf` è configurato per implementare un server di stampa accessibile a tutti gli utenti, dove però la gestione delle stampanti è affidata a un gruppo definito di amministratori, precisati alla voce `@ntadmins`.

```
[printers]
comment = All Printers
path = /var/spool/samba/
printer = HP_DeskJet_6122
browseable = no
public = yes
guest ok = yes
writable = no
printable = yes
printer admin = @ntadmins

[print$]
comment = Printer Drivers Share
path = /usr/share/cups/drivers
write list = tizio, caio, sempronio
printer admin = tizio, caio
```

La condivisione `print$` contiene le unità della stampante, alle quali i client possono accedere se le stesse non sono disponibili in modo locale. La condivisione `print$` è opzionale e potrebbe non essere necessaria a seconda dell'organizzazione. Impostando `browseable = yes`, si abilita la stampante da visualizzare nelle risorse di rete di Windows.

Server di stampa sicuro

Nell'esempio che segue il file `/etc/samba/smb.conf` è configurato per implementare un server di stampa di lettura/scrittura sicuro. Sappiamo già che impostando la direttiva `security = user`, si forza Samba all'autenticazione dei collegamenti client. Qui la sicurezza è offerta dalla mancanza della direttiva `guest = ok`, che esclude l'accesso anonimo, dalla maschera posta a 600, e infine dal fatto che i client possono usare esclusivamente driver propri, senza ricorrere al server, come si evince dalle direttive `disable spools = yes` e `use client driver = yes`.

```
[global]
.
.
.
disable spools = yes
show add printer wizard = no

[printers]
comment = All Printers
path = /var/spool/samba
printer admin = john, ed, @admins
create mask = 0600
printable = Yes
use client driver = Yes
browseable = Yes
```

16.7.3 Stampare da Windows con stampanti installate su sistemi GNU/Linux

Si presuppone che la stampante sia stata felicemente configurata con CUPS nel server di stampa GNU/Linux e anche in `/etc/samba/smb.conf`. A questo punto non resta che impostare la macchina client Windows. Sarebbe possibile farle usare dei driver di Linux, ma qui ipotizziamo di usare i driver originali per Windows della stampante. Abbiamo due possibilità per configurarla.

La prima è installare i driver con l'utilità della stampante, se ne siamo in possesso. Dal momento che tali utilità sono estremamente varie, è impossibile fornire indicazioni anche solo di massima. Tuttavia, quando l'utilità domanderà in che porta è installata la stampante, dovrebbe offrire tra le opzioni anche qualcosa come «connessione mediante rete» o «condivisione attraverso un altro computer della rete» o ancora «installazione del client per la stampa client-server» e simili. Andrà poi opportunamente inserito l'indirizzo della stampante da condividere, quando viene richiesto qualcosa come «Percorso di rete o nome coda», nella forma `\\server\nome_stampante`, dove il server di solito è il nome (ad esempio `\\darkstar`), più raramente il suo indirizzo IP (ad esempio, `\\192.168.0.2`) e il nome della stampante è quello che abbiamo indicato durante l'installazione in CUPS. Non ci si deve preoccupare nel caso compaia un messaggio che lamenta l'impossibilità, da parte del sistema di stampa, di supportare la stampante selezionata. Terminata comunque l'installazione dei driver, la stampante dovrebbe già essere disponibile in una porta locale di Windows.

Se, come a volte succede, l'utilità di installazione dovesse causare dei problemi (ad esempio, non creare la porta necessaria), sarà sempre possibile installare la stampante in locale. Quindi, andando in Pannello di controllo – Stampanti e fax e aprendo col destro del mouse la finestra Proprietà *nome_stampante*, nella linguetta Porte si creerà una nuova porta col tasto **Aggiungi porta**. Da una nuova finestra, **Porte stampanti**, basterà scegliere la voce **Local port** e premere il tasto **Nuova porta**, quindi inserire l'indirizzo come specificato sopra, e infine sceglierla come default per la stampante.

L'alternativa consiste nel valersi dell'utilità di Windows per l'installazione delle stampanti. In Pannello di controllo – Stampanti e fax, scegliere la voce **Aggiungi stampante**. Si apre il wizard **Installazione guidata stampante**. Nella finestra **Stampante locale o di rete** scegliere **Stampante di rete o stampante collegata a un altro computer**. Nella finestra successiva, **Specificare una stampante**, scegliere la voce **Connetti alla stampante** e scriverne l'indirizzo di rete alla voce: **Nome**, con la sintassi `\\server\stampante` (ad esempio, `\\darkstar\HP_DeskJet_6122`, il nome che abbiamo attribuito alla stampante in CUPS). Vanno quindi installati i driver della stampante, come ci avverte una nuova finestra; scegliere il driver adatto (quello della stampante se presente, oppure uno compatibile) dall'elenco che compare o da un floppy, e la configurazione sarà conclusa, dopo di che si potrà stampare una pagina di prova.

L'icona della stampante di rete è visibile in **Risorse di rete – Visualizza computer del gruppo di lavoro**. Qui si troveranno le icone dei vari computer appartenenti al gruppo di lavoro. Cliccando sull'icona che li rappresenta, fra le risorse condivise troveremo appunto la stampante. Cliccando ancora sull'icona di quest'ultima, si aprirà la procedura per l'installazione dei driver.

C'è infine la possibilità di installare la stampante in una porta TCP/IP, ma se e solo se la stampante stessa include un dispositivo Ethernet. Configurata la stampante a seconda del modello (alcune sono configurabili via browser, altre consentono di effettuare la stessa operazione mediante l'utilità di configurazione), per attribuirle un indirizzo IP compatibile con quelli adottati nel sistema, si inserirà tale indirizzo nel wizard di Windows che crea una nuova porta.

16.7.4 Stampare da GNU/Linux con stampanti installate su sistemi Windows

Si presuppone che la stampante sia stata regolarmente installata in Windows. C'è soltanto una modifica da apportare o un parametro da aggiungere. Nella finestra delle impostazioni per stampanti e fax (raggiungibile sia dal menu **Start** sia dal **Pannello di controllo**, selezionare la stampante col tasto destro del mouse; apparirà una finestra intitolata **Proprietà nome_stampante**. Andare nella linguetta **Condivisione**, selezionare la voce **Condividi la stampante** e scegliere un nome a piacere per **Nome condivisione**. Per motivi di compatibilità, la stringa può essere lunga al massimo otto caratteri.

In Slackware, aprire CUPS al solito indirizzo da browser, scegliere la voce **Add printer**, inserire un nome (a piacere), la collocazione (qualcosa come «remote») e una eventuale descrizione. Nella finestra successiva, alla voce **Device** scegliere dal menu a tendina la voce **Windows Printer via Samba**. Nella finestra successiva, alla voce **Device URI** inserire l'indirizzo con una sintassi (variabile a seconda delle impostazioni presenti in `/etc/samba/smb.conf`) del tipo:

```
smb://server/nome_risorsa
smb://workgroup/server/nome_risorsa
smb://utente:password@server/nome_risorsa
smb://utente:password@server/workgroup/nome_risorsa
```

dove l'utente e la password sono quelli di Windows, il server è il nome del computer Windows, il workgroup è quello di Windows riportato in `/etc/samba/smb.conf` e infine `nome_risorsa` è il nome attribuito in Windows alla stampante in **Nome condivisione**. Quindi scegliere la marca della stampante e il driver adatto, come secondo la normale procedura. Alla fine, la stampante sarà stata installata e sarà possibile stampare una pagina di prova.

Dalla versione 2.2 di Samba in poi si ha anche la possibilità di esportare da GNU/Linux ai client Windows i driver necessari per la stampa. A tale scopo viene in aiuto il comando `cupsaddsmb`, che

consente per l'appunto di esportare i driver di CUPS. Per la scelta dei driver, rimandiamo all'ampia pagina man di cupsaddsmb.

Una volta scaricati i driver e copiati nella directory `/drivers` all'interno della directory radice che di norma si trova in `/usr/share/cups` si configura `/etc/samba/smb.conf` aggiungendo la condivisione `[print$]` nel modo che segue:

```
[print$]
    comment = Printer Drivers
    path = /usr/share/cups/drivers
    write list = root
    create mask = 0664
    directory mask = 0775
```

A questo punto, avendo configurato le condivisioni `[printers]` e `[print$]`, si può usare il comando `cupsaddsmb` per esportare i drivers, che li aggiungerà a tutte le code di stampa per l'utente `root`, nella forma: `cupsaddsmb -U root -a`.

L'opzione `-a` serve a specificare tutte le code configurate nel sistema CUPS. Questo comando permette anche di aggiungere coda per coda in modo di permettere una configurazione fine della stampa remota per i client Windows.

16.8 La configurazione di Samba mediante SWAT

16.8.1 Abilitare SWAT

Samba include una utilità grafica di configurazione, SWAT (acronimo di «Samba Web Administration Tool»), che consente di impostare tutte le opzioni attraverso un browser. In tal modo risultano semplificate innumerevoli operazioni come l'arresto e il riavvio di Samba, l'impostazione dei dispositivi di stampa, la definizione dei permessi di accesso, la creazione di utenti, password e directory condivise.

Esistono anche altre interfacce grafiche (come Webmin; lo stesso KDE ne include una in Centro di Controllo – Internet e Rete – Samba), ma SWAT presenta il duplice vantaggio di essere integrato in Samba e di contenere tutte le opzioni di configurazione possibili.

Per lanciare SWAT è necessario verificare che nel file `/etc/services` sia decommentata la riga:

```
swat      901/tcp      # samba web configuration tool
```

ed eventualmente decommentarla, e che nel file `/etc/inetd.conf` allo stesso modo sia decommentata la riga:

```
swat      stream      tcp          nowait.400  root        /usr/sbin/swat      swat
```

ed eventualmente decommentarla. In Slackware la prima è già essere abilitata, la seconda invece no e dunque richiede una piccola modifica manuale (dopo la quale il demone `inetd` deve essere fatto rileggere dal sistema col comando: `/etc/rc.d/rc.inetd restart`).

A questo punto si è pronti a entrare in SWAT da un qualsiasi browser all'indirizzo `http://localhost:901`. La configurazione di SWAT sovrascrive l'originale `/etc/samba/smb.conf` eventualmente presente, eliminando tutte le parti non configurate. Se pertanto vogliamo preservarlo comunque, in vista di ulteriori cambiamenti, facciamone una copia prima di avviare la configurazione con SWAT col comando: `cp /etc/samba/smb.conf /etc/samba/smb.conf.bak`.

Va ricordato che SWAT invia le informazioni in chiaro, non supportando la crittografia. Se si configura da localhost, ciò non dovrebbe presentare problemi di sicurezza, che tuttavia potrebbero sorgere se si opera via rete. In questo caso, si può ricorrere a uno strumento di tunnelling SSL chiamato **stunnel**.

16.8.2 L'interfaccia di SWAT

La pagina di avvio presenta i link a tutta la documentazione, costituita dalle pagine man e da alcuni ponderosi How-To. Nella parte più alta della finestra, otto pulsanti consentono di impostare i vari aspetti della configurazione. Per alcuni di essi si può scegliere fra una vista semplice, che presenta solo le opzioni più importanti, e una vista avanzata, che invece presenta tutte le opzioni.

Home

Fa tornare alla pagina di avvio.

Globals

Imposta i parametri della sezione `[global]`, che dunque determinano il comportamento generale di Samba. Alla luce di quanto è stato analizzato sopra, per una configurazione di base è sufficiente inserire nella categoria **Opzioni basilari** il nome corretto alla voce `workgroup` e nella categoria **Opzioni di sicurezza** il metodo prescelto, `share` oppure – è attualmente il default – `user`. Per il resto, si possono tranquillamente accettare le opzioni proposte.

Shares

Imposta le condivisioni. Il menu **Scegli condivisione** presenta soltanto la voce `homes`. Altre directory da condividere possono essere aggiunte nel riquadro **Crea condivisione**, inserendo il loro percorso assoluto nel file system e premendo quindi il pulsante **Crea condivisione** per rendere stabili le modifiche apportate.

Printers

Configura le stampanti già presenti nel sistema (e configurate con CUPS). Si può optare per una gestione globale, o scegliere singolarmente le stampanti.

Wizard

Semplifica la gestione di `/etc/samba/smb.conf`, eliminando tutte le righe disabilitate e i commenti e lasciando perciò solo le direttive efficaci.

Status

Mostra lo stato attuale dei demoni `smbd`, `nmbd` e `winbindd`, riguardo ai client connessi, alle condivisioni attive e ai processi in corso. La pagina si aggiorna automaticamente ogni trenta secondi (ma questo valore può essere modificato a piacere). Inoltre, consente di attivare o disattivare Samba e di ricaricare il file di configurazione (nel caso vi siano state apportate delle modifiche).

View

Mostra il file `/etc/samba/smb.conf` che si è ottenuto, man mano che si apportano le modifiche mediante SWAT.

Password

Gestisce le password crittografate, e permette di aggiungere, rimuovere, abilitare e disabilitare gli utenti Samba.

Capitolo 17

La console

17.1 Tipi di console

In GNU/Linux, tutto ciò che viene fatto dalle interfacce grafiche lo si può fare da riga di comando, la quale però spesso si mostra più potente e veloce delle GUI, più ricca di opzioni e meno soggetta a eventuali crash. Pertanto la console è e resta lo strumento principe per operare in GNU/Linux. Chi sa effettuare le operazioni da console può non preoccuparsi troppo se, in un determinato computer, non c'è lo Window Manager a cui è avvezzo, perché sarà sempre e comunque in grado di lavorare col sistema.

La console, che presenta diverse interfacce (terminale, applicazione a finestra all'interno di X, terminale remoto) è lo strumento grazie a cui l'utente comunica con il sistema. Non a caso l'attivazione di un programma che gestisce il terminale è la fase conclusiva del caricamento del sistema stesso e precede immediatamente l'avvio della procedura di accesso da parte degli utenti. È questa la shell di login, in quanto chiede il nome utente e la password per entrare nel sistema. Esistono quindi altri due tipi di shell: interattive, che permettono di inserire comandi, e non interattive, ad esempio quelle che si avviano quando viene lanciato uno script.

In GNU/Linux c'è una sola console principale, che però può essere attivata attraverso innumerevoli console virtuali: le console virtuali supportate sono sessantatre, ma solo le prime sei sono configurate per l'utilizzo effettivo. La console principale di fatto non si rileva, perché non vi corrisponde nessun dispositivo fisico e vi si accede sempre mediante una o più console virtuali. Ecco perché il dispositivo `/dev/console` si incarna nella molteplicità dei dispositivi `/dev/tty` (la sigla deriva da *teletype*, un terminale molto diffuso nei primi sistemi Unix; nel 1961 ne venne realizzato un modello interamente meccanico che fungeva da telescrivente adottando il codice ascii), che vengono definiti nel file `/etc/inittab` e si correlano ai dispositivi `/dev/vc`.

17.2 Commutazione fra console

Effettuato il login, si utilizza la prima console virtuale (`tty1`), e quindi è possibile commutare verso altre console, connettendosi diverse volte, come utenti diversi (connettersi a più console con lo stesso utente può generare dei problemi), aprendo innumerevoli sessioni. Da console, la commutazione si effettua col comando **Alt+F1**, ..., **Alt+F6**, mentre con **Alt+F1** si ritorna alla console originaria; in alternativa, si può usare la combinazione **Alt+↑** per passare alla console successiva e **Alt+↓** per passare a quella precedente. Dall'interno del server X, invece, il comando è **Ctrl+Alt+F1**, ..., **Ctrl+Alt+F6**, mentre **F7** è la console dove di default è attivato il server X.

In alternativa alla pressione di determinati tasti, la commutazione può essere effettuata (ma solo da root) col comando: `chvt n`, dove `n` è il numero della console a cui si vuole commutare.

Per conoscere in quale console si è in un certo momento, si utilizza il comando: `fgconsole`.

17.3 Prompt e comandi principali

Il prompt della console (e così la shell `bash`) si presenta, ad esempio, `samiel@darkstar:~$` se appartiene a un utente, `root@darkstar:~#` se appartiene a `root`. La sua sintassi richiede: comando [opzioni] [argomenti]; di norma, le opzioni sono precedute da un trattino semplice, a volte da due trattini. Molti dei comandi di digitazione richiamano quelli usati dell'editor Emacs. Ad esempio, per spostarsi a destra di un carattere si può usare la freccetta, ma anche digitare `Ctrl+f`, mentre per spostarsi a sinistra, oltre alla freccetta, `Ctrl+b`; per cancellare il carattere che precede il cursore si può usare il tasto **Canc**, ma anche la combinazione `Ctrl+d`, e via dicendo. È comunque possibile adottare i comandi classici di **Vi** in luogo di quelli di **Emacs** lanciando il comando: `set -o vi`.

Riportiamo un elenco dei comandi principali:

<code>Ctrl+b</code> o <code>←</code>	= sposta il cursore di un carattere a sinistra
<code>Ctrl+f</code> o <code>→</code>	= sposta il cursore di un carattere a destra
<code>Ctrl+a</code>	= sposta il cursore all'inizio della riga
<code>End</code> o <code>Ctrl+e</code>	= sposta il cursore alla fine della riga
<code>Alt+f</code>	= sposta il cursore avanti di una parola
<code>Alt+b</code>	= sposta il cursore indietro di una parola
<code>Canc</code> o <code>Ctrl+d</code>	= cancella il carattere dov'è posizionato il cursore
<code>Backspace</code> o <code>Ctrl+h</code>	= cancella il carattere che precede il cursore
<code>Alt+d</code>	= cancella la parola da dov'è posizionato il cursore
<code>Ctrl+w</code>	= cancella la parola che precede il cursore
<code>Ctrl+k</code>	= cancella la riga dalla posizione del cursore alla fine
<code>Ctrl+u</code>	= cancella la riga dov'è posizionato il cursore
<code>Ctrl+y</code>	= incolla l'ultimo testo cancellato
<code>Ctrl+t</code>	= inverte gli ultimi due caratteri a sinistra del cursore
<code>Alt+t</code>	= inverte le ultime due parole a sinistra del cursore
<code>Alt+u</code>	= trasforma in maiuscolo una parola a partire da dov'è il cursore
<code>Alt+l</code>	= trasforma in minuscolo una parola a partire da dov'è il cursore
<code>Maiusc+Pag↑</code>	= fa scorrere il testo di una schermata verso l'alto
<code>Maiusc+Pag↓</code>	= fa scorrere il testo di una schermata verso il basso
<code>Ctrl+l</code>	= pulisce la console
<code>Ctrl+n</code> o <code>↑</code>	= sposta il cursore sul comando successivo della history
<code>Ctrl+p</code> o <code>↓</code>	= sposta il cursore sul comando precedente della history
<code>Esc<</code>	= sposta il cursore sul primo comando della history
<code>!!</code>	= riesegue l'ultimo comando
<code>!n</code>	= richiama il comando corrispondente al numero <i>n</i> nella history (dove <i>n</i> è il numero del comando)
<code>!<i>stringa</i></code>	= richiama dalla history il comando che inizia coi caratteri specificati (dove <i>stringa</i> sono i caratteri con cui inizia un comando)
<code>Esc</code> o <code>Tab</code>	= richiama la funzionalità di completamento

Se si devono indicare più comandi di seguito, essi vanno separati col punto e virgola (;).

Se si devono utilizzare più righe prima di dare il comando con **Invio**, il carattere di barra inversa (`\`) consente di ignorare il codice del tasto **Invio**, per cui il testo è interpretato come in continuazione sulla riga successiva.

Un comando che impiega parecchio tempo per essere eseguito, e che perciò terrebbe occupata la console, può essere messo in background facendolo terminare col simbolo della e commerciale (&), in modo da rendere nuovamente disponibile la shell.

La console può essere pulita col comando `clear` e reinizializzata col comando `reset`.

Per uscire da una console, si dà il comando `logout`, `exit` oppure anche `Crtl+d`.

La console offre una funzionalità di completamento che si richiama con la combinazione **Esc+Tab** (o solo **Tab**). Se esistono più comandi o nomi di file che iniziano per quella lettera o per quella stringa, la console li visualizza tutti, in attesa che vengano specificate lettere ulteriori, per restringere il campo delle possibilità fino ad arrivare a una scelta univoca. Se si digita **Esc+**, la shell mostra tutti i nomi di file che corrispondono a quella stringa incompleta.

17.4 La history

La console ha una funzione, chiamata *history*, che tiene traccia dei comandi precedentemente digitati: ciò consente di richiamarli senza doverli digitare di nuovo per intero. Il comando `history` restituisce i comandi digitati col numero che li identifica, in ordine cronologico. Si può risalire o ridiscendere fra i comandi con i tasti freccetta, mentre con `Pag↑` e `Pag↓` si va rispettivamente al primo e all'ultimo comando; si può facilitarne la consultazione reindirigendo il comando a `more` o `less`. Si possono anche richiamare i comandi con `!numero_comando`. L'elenco dei comandi viene registrato nel file `~/.bash_history`, impostato di default a 500. Il comando `echo $HISTFILE` visualizza la posizione del file di elenco, mentre il comando `echo $HISTSIZE` visualizza il valore numerico della variabile. Quest'ultima può essere modificata aggiungendo in `~/.bashrc` una riga del tipo:

```
export HISTSIZE=nn
```

dove *nn* è un numero a piacere che indica il nuovo numero dei comandi di cui `history` tiene traccia. Dopo di ciò, si fa rileggere `~/.bashrc` dal sistema col comando `source`, e la nuova configurazione risulta abilitata. La `history` può essere ripulita col comando `history -c`.

17.5 Il server X in più terminali

È possibile lanciare più server X, uno per ciascuna console virtuale, ma allora il comando non sarà semplicemente il classico `startx` (infatti in questo caso si riceverebbe l'avviso: «Fatal server error: Server is already active for display 0»), bensì `startx -- :n`, dove *n* è il numero della console 2, ..., 6, a seconda di quante sessioni grafiche sono aperte. Per verificare a quale console siamo connessi, daremo il comando `tty`. Se abbiamo dei dubbi su quale utente è connesso a un certo terminale, daremo il comando `whoami`.

Tuttavia, rispetto all'avvio in modalità testuale (runlevel 3) e al successivo passaggio alla modalità grafica tramite `startx`, l'avvio diretto in modalità grafica (runlevel 4) fa perdere in Slackware la possibilità di commutare tramite **Ctrl+Alt+Fx** (con $x=2-5$) le console virtuali dall'interno del server X (in questo caso appare una schermata nera senza prompt e per ritornare alla modalità grafica è necessario digitare **Ctrl+Alt+F7**). Si potranno raggiungere solo la console «reale» (**F1**) e un'altra console (**F6**), e non le altre (2-4). Ciò è dovuto al fatto che al runlevel 4 Slackware attiva di default solo la console 6 e disattiva le altre 2-5. Per ripristinare la possibilità della commutazione è sufficiente modificare la sezione sotto riportata del file `/etc/inittab`:

```
# These are the standard console login getties in multiuser mode:  
c1:1235:respawn:/sbin/agetty 38400 tty1 linux
```

```
c2:1235:respawn:/sbin/agetty 38400 tty2 linux
c3:1235:respawn:/sbin/agetty 38400 tty3 linux
c4:1235:respawn:/sbin/agetty 38400 tty4 linux
c5:1235:respawn:/sbin/agetty 38400 tty5 linux
c6:12345:respawn:/sbin/agetty 38400 tty6 linux
```

aggiungendo 4 alle console da 1 a 5 che interessa attivare, per cui avremo (se intendiamo ripristinarle tutte):

```
# These are the standard console login getties in multiuser mode:
c1:12345:respawn:/sbin/agetty 38400 tty1 linux
c2:12345:respawn:/sbin/agetty 38400 tty2 linux
c3:12345:respawn:/sbin/agetty 38400 tty3 linux
c4:12345:respawn:/sbin/agetty 38400 tty4 linux
c5:12345:respawn:/sbin/agetty 38400 tty5 linux
c6:12345:respawn:/sbin/agetty 38400 tty6 linux
```

Scheda n. 26

Inizializzazione e runlevel

Slackware ha una modalità del tutto particolare nel gestire l'inizializzazione del sistema, ossia la gerarchia dei programmi che fanno poi funzionare il sistema nel suo complesso. Si tratta del cosiddetto stile BDS, laddove la più gran parte delle distribuzioni GNU/Linux ricorre invece al System V. Entrambi gli stili si basano su script di shell e non su programmi. La differenza fondamentale che li contraddistingue consiste nel fatto che nel System V ogni runlevel ha una sua subdirectory con gli script di inizializzazione, mentre nello stile BDS è presente un solo script per ogni runlevel.

Stile SysV e stile BDS

Gli script SysV prendono argomenti del tipo `start`, `stop`, `restart` e simili, a seconda del comando da passare. Perciò si usa, ad esempio, il comando `/etc/initrc.d/bind start` per far partire Bind, e `/etc/rc.d/bind stop` per fermarlo. Inoltre, gli init in stile SysV fanno spesso ricorso a collegamenti simbolici per organizzare il processo di avvio: di conseguenza in `/etc/initrc.d/rc.4` potrebbero essere presenti dei collegamenti simbolici a script di directory. Il nome dei link simbolici è, ad esempio, `S10network`, `S25xdm`, ecc., dove S sta per «start» (che determina l'avvio del servizio), mentre K sta per «kill» (che blocca o «uccide» il processo), mentre i numeri indicano l'ordine nel quale devono partire. Il principale vantaggio dell'init in stile SysV è che esso è in grado di configurare «automagicamente» diversi aspetti dei processi. Ad esempio, nel runlevel 6 può esserci un link simbolico in `/etc/rc.d/rc.6` chiamato `K75bind`, che blocca Bind se il file a cui è collegato è configurato per far ciò.

Il principale svantaggio dello stile SysV sta nella sua notevole complessità. Per aggiungere un qualche servizio, è necessario redigere uno script in stile SysV che gestisca quanto meno `start` (e possibilmente anche `stop`). Inoltre, bisogna impostare correttamente i collegamenti simbolici giusti a ogni runlevel nel quale lo script dev'essere eseguito; se poi esso dev'essere eseguito fra due script con numero sequenziale, si dovranno rinumerare progressivamente i link simbolici (se lo script z dev'essere eseguito fra gli script `S10x` e `S11y`, devo attribuire allo script z il numero `S11z` e rinumerare `S11z` come `S12z`).

Infine, è piuttosto scomodo modificare temporaneamente il processo di boot di SysV, se ad esempio si vuole che un determinato servizio non parta al boot successivo. Il modo più facile è rinominare il link

simbolico, ma può doversi fare per ogni runlevel, e in caso di ripristino della situazione iniziale bisognerà ricreare manualmente tutti i link. Per evitare questo complesso gioco, Slackware ricorre a script in stile BDS. Essi sono degli script di shell che si avviano sequenzialmente e non prendono argomenti come «start» o «stop»: semplicemente, partono quando il sistema entra nel loro runlevel.

Il principale svantaggio degli script in stile BDS sta nelle procedure diverse che si devono volta per volta adottare per controllare i demoni. Ad esempio, se voglio bloccare («uccidere») Bind, devo individuare il PID del processo named col comando: `ps aux|grep named` e quindi ucciderlo, ma non posso ricorrere a `/etc/init.d/bind stop`, a meno che non abbia redatto uno script in stile SysV per farlo.

Il principale vantaggio degli script in stile BDS è che sono facili da comprendere e modificare (con un qualsiasi editor di testo). Ad esempio, per avviare un servizio z, è sufficiente aggiungere la riga `/usr/local/bin/z` al file `/etc/rc.d/rc.local`, e il servizio stesso verrà eseguito a tutti i runlevel dove viene eseguito `rc.local`. Se voglio che quel servizio sia eseguito solo nel runlevel 4, lo aggiungerò solo in `/etc/rc.d/rc.4`, che non è una directory, bensì uno script. Se voglio modificare l'ordine di esecuzione degli script, è sufficiente che ne modifichi l'ordinamento nello script che li lancia. Infine, per escludere un servizio basta commentare la chiamata, e decommentarla poi quando serve.

Per quanto non sia difficile passare da uno stile all'altro (basta copiare da un sistema all'altro il file `inittab` e i file `.rc`), c'è chi sostiene la maggiore potenza del SysV e chi invece la semplicità del SysBDS.

Inizializzazione del sistema

In Slackware, subito dopo l'esecuzione del kernel, viene eseguito lo script `init`, che legge il file `/etc/inittab` per scegliere la modalità di avvio (cioè il runlevel). Viene quindi eseguito lo script `/etc/rc.s/rc.S` che appronta il sistema, prima di entrare effettivamente nel livello (runlevel) desiderato. Questo script compie numerose operazioni: abilita la memoria virtuale, monta il file system, pulisce alcune directory di log, inizializza le periferiche plug&play, carica i moduli del kernel, configura eventuali periferiche PCMCIA, imposta le porte seriali e carica gli script System V, qualora ve ne siano. Inoltre invoca altri script presenti in `/etc/rc.d` che completano il processo di inizializzazione configurando altre periferiche: `/etc/rc.d/rc.modules` (che carica i moduli del kernel ed eventualmente invoca lo script `/etc/rc.d/rc.netdevice` per configurare la rete); `/etc/rc.d/rc.pcmcia` (che configura le periferiche di questo tipo qualora siano presenti); `/etc/rc.d/rc.serial` (che configura le porte seriali); `/etc/rc.d/rc.sysvinit` (che cerca ed eventualmente esegue gli script per il runlevel desiderato).

I runlevel

Il runlevel descrive lo stato in cui girerà il sistema, che può essere monoutente, multiutente, con o senza i servizi di rete, grafico o testuale. I runlevel sono i seguenti:

runlevel 0 arresta il sistema

runlevel 1 avvia il sistema in modalità monoutente

runlevel 2 avvia il sistema in modalità multiutente

runlevel 3 avvia il sistema in modalità multiutente in modalità testuale (è il livello predefinito)

runlevel 4 avvia il sistema in modalità multiutente e grafica (cioè all'interno del server X con KDM, XDM ed eventualmente GDM)

runlevel 5 non viene usato (e può essere personalizzato)

runlevel 6 riavvia il sistema

I runlevel sono gestiti da script, secondo la tabella che segue:

RUNLEVEL	SCRIPT
Runlevel 0	/etc/rc.d/rc.0 (che in realtà è un semplice link simbolico a rc.6)
Runlevel 1	/etc/rc.d/rc.K
Runlevel 2	/etc/rc.d/rc.M
Runlevel 3	/etc/rc.d/rc.M
Runlevel 4	/etc/rc.d/rc.4
Runlevel 5	–
Runlevel 6	/etc/rc.d/rc.6

Il runlevel di default in Slackware è 3; se si vuole entrare direttamente in modalità grafica (come abbiamo visto sopra) si deve passare al runlevel 4, modificando il file `/etc/inittab`.

L'inizializzazione della rete

L'inizializzazione della rete (attuata nei runlevel 2, 3 e 4) richiede altri script: `/etc/rc.d/rc.inet1` (file configurato da `netconfig`, responsabile della configurazione della rete); `/etc/rc.d/rc.inet2` (eseguito dopo `/etc/rc.d/rc.inet1`, fa partire i servizi di base della rete); `/etc/rc.d/rc.atalk` (avvia i servizi AppleTalk); `/etc/rc.d/rc.httpd` (avvia il server web Apache); `/etc/rc.d/rc.samba` (avvia Samba, cioè il servizio di condivisione file e stampanti per Windows); `/etc/rc.d/rc.news` (avvia il server delle news).

Il System V

Slackware è comunque compatibile con il System V. Lo script `/etc/rc.d/rc.sysvinit` cerca eventuali script System V presenti in `/etc/rc.d` per eseguirli quando il runlevel sia appropriato. Ciò torna utile in quanto alcuni programmi, per poter funzionare, installano script di questo tipo, che dunque il sistema dev'essere in grado di gestire.

Altri script

Nella directory `/etc/rc.d` esistono svariati altri script, invocati da quelli principali, che gestiscono altri aspetti del sistema, ad esempio `/etc/rc.d/rc.alsa` (che abilita il server sonoro), `/etc/rc.d/rc.cups` (che gestisce le stampanti), `/etc/rc.d/rc.font` (che carica il font scelto per la console), `/etc/rc.d/rc.gdm` (che abilita l'uso del mouse in console), `/etc/rc.d/rc.httpd` (che avvia Apache) e innumerevoli altri. Lo script `/etc/rc.d/rc.local` di default è vuoto, e viene eseguito per ultimo; serve per aggiungere altri comandi definiti da root che vanno lanciati automaticamente durante il boot del sistema. Tutti gli script si possono modificare con un semplice editor di testo. Essi devono essere eseguibili; per evitarne l'esecuzione, va eliminato il permesso di esecuzione col comando: `chmod a-x`.

17.6 L'avvio del sistema

Di default, con Slackware il sistema parte in modalità testuale e non grafica. Dopo il caricamento, esso si blocca al prompt (cioè alla console di login), col vantaggio che possiamo scegliere se entrare come root oppure come utente in modalità grafica, o ancora se desideriamo effettuare alcune operazioni da console o con altre applicazioni che non richiedono un desktop manager (editor come Emacs o vi, gestori di posta come mutt o pine, un esecutore di MP3 come mpg321 e via dicendo).

17.7 Modificare il manager di avvio

Per modificare il manager caricato da `startx` si può lanciare il comando `xwmconfig` e scegliere lo Window Manager che si preferisce. In alternativa, lo stesso risultato può essere ottenuto modificando a mano i file responsabili dell'avvio del server X. Dopo aver identificato in `/etc/X11/xinit` la serie di `xinitrc.nome`, dove `nome` sta per il nome dei Window Manager installati, è sufficiente copiare quello pre-

scelto in `/home` come `.xinitrc`. In `/etc/X11/xinit` sono contenuti tutti gli script di avvio di ogni Window Manager disponibile. Ciò si spiega perché, quando `startx` fa partire `initrc`, il suo file di configurazione determina quali programmi (in particolare, il gestore di finestre) debbano essere lanciati all'avvio di X. Per prima cosa, `xinit` controlla se esiste un file `.xinitrc` nella home dell'utente e, nel caso lo trovi, lo fa eseguire; altrimenti usa il file predefinito, che si trova in `/var/X11R6/lib/xinit/xinitrc`. Se si hanno utenti diversi nel sistema, i quali intendono adottare window manager diversi, basta creare nella `/home` di ciascuno un link simbolico allo script `xinitrc` prescelto. Se l'utente A vuole avere KDE come Window Manager predefinito e invece l'utente B GNOME, allora il primo creerà un link in questo modo: `ln -s /etc/X11/xinit/xinitrc.kde /home/A/.xinitrc`, mentre il secondo in questo modo: `ln -s /etc/X11/xinit/xinitrc.gnome /home/B/.xinitrc`. L'operazione può essere effettuata dagli utenti stessi (e non da root) all'interno della loro `/home`, digitando ad esempio il comando:

```
ln -s /etc/X11/xinit/xinitrc.blackbox ~/.xinitrc.
```

17.8 Avviare automaticamente in modalità grafica

Se invece desideriamo entrare automaticamente in modalità grafica e avviare il nostro desktop manager di default (nel nostro caso, KDE) senza dover digitare nulla, dobbiamo editare il file `/etc/inittab` e modificare la sezione:

```
# Default runlevel. (Do not set to 0 or 6)
id:3:initdefault:
```

in:

```
# Default runlevel. (Do not set to 0 or 6)
id:4:initdefault:
```

Infatti, «3» indica la modalità di avvio testuale e «4» la modalità grafica. In questo caso viene invocato `/etc/rc.d/rc.4`, che segue un preciso ordine di caricamento: dapprima `kdm` (KDE) e quindi `xdm` (un display manager di base che consente soltanto di effettuare una registrazione nel sistema). Se si installasse anche GNOME, allora questo verrebbe caricato per primo. Poiché l'utilità `xwmconfig` non modifica questo stato di cose, se si vuole caricare KDE in luogo di GNOME bisogna intervenire manualmente, rieditando `/etc/rc.d/rc.4` in modo da invertire la gerarchia degli accessi. Perciò le righe:

```
# Try to use GNOME's gdm session manager:
if [ -x /usr/bin/gdm ]; then
    exec /usr/bin/gdm -nodaemon
fi
# Not there!
# OK, try to use KDE's kdm session manager:
if [ -x /opt/kde/bin/kdm ]; then
    exec /opt/kde/bin/kdm -nodaemon
fi
```

andranno riscritte come segue:

```
# Try to use KDE's kdm session manager:
if [ -x /opt/kde/bin/kdm ]; then
    exec /opt/kde/bin/kdm -nodaemon
fi
# Not there!
# OK, try to use GNOME's gdm session manager:
if [ -x /usr/bin/gdm ]; then
    exec /usr/bin/gdm -nodaemon
fi
```

In alternativa, è sufficiente commentare nello stesso file le righe relative a Gnome. Una terza soluzione, la più semplice ed elegante, consiste nel togliere a gdm l'eseguibilità col comando: `chmod a-x /usr/bin/gdm`. Eventuali altre piccole modifiche, riguardanti l'ingresso in KDE, possono essere effettuate dal **Centro di Controllo** di KDE stesso. Se il sistema è impostato al runlevel 3, è sempre possibile avviare automaticamente la modalità grafica passando al boot l'opzione `telinit 4`. Quando appare Lilo, sulla riga che mostra l'avvio di Slackware (di default l'indicazione è «Linux») basta aggiungere di seguito l'opzione indicata. Allo stesso modo, se il runlevel predefinito è 4, per evitare che si avvii automaticamente il server X e il Desktop Manager prescelto e per rimanere invece al prompt della console, basta aggiungere all'avvio l'opzione `telinit 3`.

KDE presenta una comoda utilità per automatizzare l'avvio grafico, consentendo inoltre di scegliere se un qualche utente farà l'ingresso senza dover digitare la password. Da **Centro di controllo – Amministrazione di sistema – Gestione degli accessi** nella scheda **Comodità**, alla voce **Abilita ingresso automatico** si sceglie il nome dell'utente da autorizzare; per velocizzare ulteriormente lo si può impostare come **Utente preselezionato** in modo che il suo nome appaia subito nel gestore degli accessi, e infine alla voce **Abilita accesso senza password** si può consentire a utenti prescelti l'ingresso senza password. In questo modo l'utente principale sarà messo nella condizione di accendere il computer e di trovarsi automaticamente nella sua home con KDE già attivato.

Capitolo 18

La shell bash

18.1 Le shell di Slackware

La shell è un prompt interattivo dei comandi. Essa costituisce una «finestra» per interagire con la console. Nei sistemi GNU/Linux ci sono vari tipi di shell (detti anche «pseudo-terminali»), di norma suddivise in sh compatibili e csh compatibili: le shell disponibili sono elencate in `/etc/shells` (e si possono visualizzare velocemente col comando `cat /etc/shells`). In Slackware sono: **bash**, **tcsh**, **csh**, **ash**, **ksh**, **zsh**. Qui ci occuperemo esclusivamente della shell bash (acronimo di «Bourne Again Shell»), la shell di GNU/Linux conforme allo standard POSIX. Posix sta per «Portable Operating System Interface for uniX», un'interfaccia standard per il sistema operativo Unix creata verso la fine degli anni Ottanta (e progressivamente aggiornata) allo scopo di uniformare le varie versioni di Unix.

Slackware fornisce diversi emulatori di terminale, come **Konsole** (per KDE), **xterm** o **rxvt**, appunto per rendere possibile la connessione alla console dal suo interno. In ogni caso, tutte le console virtuali (**xterm** e **remote**) fanno capo a `/dev/pts/*` (si può constatare facilmente come, se è attivo `udev`, i dispositivi `-0`, `-1` ecc. aumentino se si aprono più shell), mentre se si usa `udev` (o `devfs` senza aliasing), anche le `tty` diventano `/dev/vc/*`.

La shell predefinita del sistema è assegnata agli utenti dall'ultimo campo delle corrispettive righe in `/etc/passwd`, mentre le impostazioni generali di base (le principali variabili d'ambiente) sono precisate in `/etc/profile`. Sebbene la shell di default nei sistemi GNU/Linux sia in genere bash, è possibile cambiare shell digitandone il nome dalla riga di comando (non così nei sistemi Unix). La shell può essere modificata col comando `chsh`. Per uscire dalla shell si usano ovviamente gli stessi comandi già indicati a proposito della console, e cioè la combinazione **Ctrl+d**, e le stringhe `logout` oppure `exit`.

18.2 Standard input e standard output

La shell ha uno standard input (**stdin**) e uno standard output (**stdout**). A livello immediato, lo standard input è la tastiera, dove si digitano i comandi, e lo standard output è lo schermo, dove essi vengono visualizzati. Ad esempio, il comando `cat /etc/fstab` visualizza a schermo il file stesso. Il semplice comando `cat`, invece, non fa tornare al prompt della shell perché resta in attesa; se si digita una qualsiasi stringa e si dà **Invio**, essa verrà ripetuta da `cat` finché non si digita `Ctrl+d` per ritornare al prompt. La riga digitata è in questo caso lo `stdin`, la riga ripetuta lo `stdout`.

I comandi di shell accettano per lo più opzioni e argomenti. Ad esempio, in `ls -la /usr/bin`, `ls` è il comando, `-la` l'opzione, `/usr/bin` l'argomento. Vediamo un esempio: il comando `ls` elenca il contenuto di una directory, nell'impostazione predefinita quella corrente; l'opzione `-l` fornisce un elenco dettagliato, l'opzione `-a` mostra anche i file (e le directory) nascosti. Se si passa come argomento l'indirizzo di

una directory diversa, il comando ne mostrerà il contenuto. Va infine ricordato che i file system tipici di Unix – e conseguentemente anche il modo di passare i comandi alla shell (a differenza del prompt del DOS) – sono *case sensitive*, cioè sensibili alla differenza fra maiuscole e minuscole.

18.3 Comandi di base della bash

La shell bash offre un vasto numero di comandi *built-in*, molti dei quali assai ricchi di ulteriori opzioni. Numerosissimi sono poi i comandi esterni. Analizziamo ora qualche comando di base, per familiarizzare con le procedure più elementari, fermo restando che per tutte le altre opzioni va consultata la rispettiva pagina man.

18.3.1 Elencare file

Per visualizzare i file di una directory, si usa il comando `ls`. Esso, di default, mostra i file e le subdirectory della directory corrente, a meno che non se ne precisi un'altra. Fra le numerosissime opzioni (una cinquantina), una delle più utili è `-l`, che mostra i dettagli sui file:

```
samiel@darkstar:~$ ls -l /usr/doc/abiword-2.0.12/
totale 57
-rw-r--r-- 1 root root 5 2002-10-01 19:45 BUILD.TXT
-rw-r--r-- 1 root root 6216 2003-03-18 08:23 BiDiReadme.txt
-rw-r--r-- 1 root root 17980 2002-04-15 15:25 COPYING
-rw-r--r-- 1 root root 557 1998-10-21 21:54 COPYRIGHT.TXT
-rw-r--r-- 1 root root 14958 2004-03-18 01:58 CREDITS.TXT
-rw-r--r-- 1 root root 1248 1999-11-16 01:11 README.TXT
drwxr-xr-x 2 root root 616 2004-10-19 08:38 docs/
```

Nella prima riga è indicato lo spazio occupato dalla directory, in blocchi da 1024 KB. Le righe successive sono organizzate in colonne.

Nella prima colonna, il trattino (-) indica che si tratta di un file, la d di una directory, mentre la l indicherebbe i link simbolici. Le lettere che seguono indicano i permessi del file.

La seconda colonna indica il numero di collegamenti fisici al file. Ogni file ha almeno un collegamento fisico, che è il nome originale del file, mentre ogni directory ne ha almeno due, e cioè il nome della directory, visibile dalla directory superiore, e il file speciale `.` (punto) situato nella directory stessa, che indica la directory corrente quando si è al suo interno.

La terza e quarta colonna indicano utente e gruppo a cui il file (o la directory) appartiene.

La quinta colonna indica le dimensioni del file in byte.

La sesta colonna indica la data e l'ora dell'ultima modifica.

La settima e ultima colonna contiene il nome del file (o della directory).

L'opzione `-a` mostra i file (e le directory) nascosti (quelli il cui nome inizia col punto).

L'opzione `-R` mostra le directory in modo ricorsivo, cioè con le loro eventuali subdirectory; se l'elenco è lungo, conviene combinarlo con `more` o `less` (nella forma `ls ... | more` oppure `ls ... | less`). Reindirizzato a `wc`, nella forma `ls | wc -l`, conta i file contenuti nella directory, anziché indicarli per nome.

L'opzione `-F` consente di distinguere i file dalle directory facendo seguire al nome di queste ultime una barra (/); inoltre dettaglia i tipi di file, ponendo un asterisco (*) dopo i file eseguibili e il simbolo «at» (@) dopo i link simbolici.

L'opzione `-t` mostra i file in ordine cronologico, partendo dai più recenti.

L'opzione `-u` mostra i file in ordine cronologico di accesso.

L'opzione `--color` mostra i file e le directory distinguendone i vari tipi con colori diversi: i nomi dei file in verde, quelli delle directory in blu, in rosso gli archivi compressi, in giallo i dispositivi e i link in azzurro. In Slackware questa impostazione è già presente in `/etc/DIR_COLORS` e nei due file `/etc/profile.d/coreutils-dircolors.*`: dunque l'opzione `--color` può non essere precisata, ma può essere ulteriormente personalizzata da ciascun utente settando dei nuovi e diversi parametri in `~/.bash_profile`.

Il comando `tree` mostra anche graficamente l'albero delle directory. Con l'opzione `-d` esso mostra unicamente i nomi delle directory.

Un altro modo per vedere i file dall'interno della shell è ricorrere a un browser, grafico oppure testuale (come `lynx`). In questo caso lo si invocherà dandogli l'indirizzo preceduto dal prefisso `file:/`. Infine si può ricorrere a uno strumento di gestione dei file come **Midnight Commander**, invocato col comando `mc`. Mentre `root` può visualizzare tutte le directory, gli utenti possono visualizzare solo quelle a cui hanno almeno l'accesso di lettura: un utente qualsiasi si vedrà negato il permesso di visualizzare la directory `/root`.

18.3.2 Copiare file e directory

Per copiare file o directory il comando di base è `cp`. La sintassi è la seguente: `cp file destinazione`.

In tutti questi casi, il file copiato sarà identico all'originale, tranne che per il *timestamp*, dal momento che riporterà come data e ora quella della sua creazione.

L'opzione `-a` (che abbrevia `--archive`) conserva nella copia quanto è possibile della struttura e degli attributi dei file originali, ma non la struttura di directory.

In generale, il file copiato avrà gli stessi permessi dell'originale e l'utente e il gruppo del copiante, modificati dall'`umask` proprio dell'utente, a meno che non si precisi l'opzione `-p`, la quale preserva gli attributi originali (proprietario, gruppo e permessi). Se però vengono copiate ricorsivamente delle directory, i permessi di quelle create *ex novo* sono quelli finali, in modo da permettere al processo di leggere, scrivere e cercare la directory appena creata.

L'opzione `-f` forza la copiatura, producendo automaticamente la sovrascrittura di eventuali file con lo stesso nome presenti nella directory di destinazione. L'opzione `-i` copia invece in maniera interattiva, chiedendo conferma per effettuare la sovrascrittura, se nella directory di destinazione è presente un file con lo stesso nome del file originale.

Per copiare contemporaneamente più file, i loro nomi vanno spaziati oppure inclusi in graffe, separati da una virgola, secondo la seguente sintassi: `cp file1,file2,...,filen destinazione`.

Per effettuare una copia ricorsiva di una directory (cioè comprendente tutte le sue subdirectory), va precisata l'opzione `-R`, che però non copia i link simbolici e non preserva gli attributi originali. Per effettuare una copia di questo tipo (cioè tale da includere i link e i permessi originali), va adottata l'opzione `-a`. Per copiare e/o rinominare un file ci si può servire anche di `cat`, reindirizzando il comando alla destinazione e al nuovo nome del file. Ad esempio, `cat /documenti_1/testo1.txt > /documenti_2/testo2.txt` copierà il file `testo1.txt` (che sarà conservato) nella directory `/documenti_2` rinominandolo `testo2.txt`.

18.3.3 Spostare e rinominare file e directory

Il comando `mv` serve sia a spostare, sia a rinominare file. Se l'argomento è costituito solo dal nome del file e dalla directory di destinazione, allora il file verrà spostato in quest'ultima. Se invece esso presenta anche un altro nome di file, il file di origine verrà rinominato ed eventualmente spostato:

```
mv file /destinazione sposta il file
```

```
mv file1 file2 rinomina il file1 come file2
```

```
mv file1 file2 /destinazione sposta file1 nella destinazione rinominandolo file2.
```

Se nella directory di destinazione esiste un file con lo stesso nome, esso viene automaticamente sovrascritto. La stessa procedura si adotta per spostare una directory. Quando si precisa la destinazione, la directory da copiare diventa una subdirectory di quella di destinazione.

In maniera del tutto analoga al comando `cp`, anche a proposito di `mv` l'opzione `-f` forza lo spostamento, sovrascrivendo sempre e senza messaggi di avviso, mentre l'opzione `-i` fa sì che il programma avverta che è già presente un file con lo stesso nome e chieda se si desidera effettivamente sovrascriverlo. Per rendere stabile questo meccanismo di sicurezza, si può creare un alias che trasformi stabilmente il comando `mv` in `mv -i`.

Si possono convertire in lettere minuscole i nomi di file scritti in maiuscole col comando `chcase`, che in realtà non è un comando della shell, ma uno script in Perl non incluso in Slackware e scaricabile da Internet. Il comando non trasforma i nomi delle directory, a meno che non si aggiunga l'opzione `-d`. Questo comando compie invece l'operazione contraria, dal minuscolo al maiuscolo, con l'opzione `-u`. Nei sistemi Unix, i nomi dei file possono essere costituiti da lettere maiuscole, minuscole e da caratteri alfanumerici. Le estensioni non sono necessarie, ma all'occorrenza possono essere tradizionali (il che favorisce l'identificazione del tipo del file), come `.txt` o `.jpeg`, oppure inventate, o anche multiple, come potrebbe essere `.txt.back`. In ogni caso, i file sono riconosciuti dalle applicazioni indipendentemente dalla presenza dell'estensione.

18.3.4 Trovare file

Ci sono quattro comandi fondamentali per la ricerca di file.

Il comando `which`

Il primo è il comando `which`, che esegue la sua ricerca all'interno del path dell'utente e restituisce la prima occorrenza del programma cercato con il relativo percorso assoluto. Se però il file non si trova nel path dell'utente o comunque in una directory a cui egli abbia accesso, esso non verrà individuato. Ad esempio, il comando `which bash` dà come output:

```
/usr/bin/bash
```

Il comando `whereis`

Il secondo è il comando `whereis`, che funziona in modo analogo a `which`, ma è in grado di cercare anche pagine di manuale e file sorgenti. Ad esempio, il comando `whereis bash` dà come output:

```
bash: /bin/bash /usr/bin/bash /usr/man/man1/bash.1.gz /usr/share/man/man1/bash.1.gz
```

Il comando `find`

Il terzo è il comando `find`, che permette di cercare qualunque cosa all'interno del sistema, da cui deriva una sua certa lentezza. Richiede però dei parametri in più, ossia almeno il path dove effettuare la ricerca (sempre che sia accessibile all'utente che effettua la ricerca) e il criterio. Quest'ultimo viene precisato mediante numerose opzioni.

Se il criterio è rappresentato dal nome del file, allora l'opzione sarà `-name`, seguita dal nome che intendiamo trovare; essa distingue tra maiuscolo e minuscolo, cosicché, se volessimo effettuare la ricerca prescindendo dalla distinzione fra maiuscole e minuscole, adotteremmo l'opzione `-iname`. Valgono qui tutti i caratteri jolly. Ad esempio, per cercare in tutto il sistema la posizione del file `bash` si usa il comando: `find / -name bash`.

L'opzione `-regex` cerca invece quei file i cui nomi corrispondono a un'espressione regolare o a un modello che descrive una serie di stringhe.

L'opzione `-size` seguita dalla dimensione (di default fissata in blocchi di 512 byte) cerca i file in relazione alla dimensione specificata. Se essa è preceduta dal segno più (+), la corrispondenza viene stabilita coi file di dimensione maggiore di quella indicata, se preceduta dal segno meno (-) coi file di dimensioni minori; se invece non si aggiunge nessun prefisso, la corrispondenza verrà cercata con i file che corrispondono esattamente alla dimensione indicata. Se si preferisce indicare la dimensione in byte, si farà seguire il numero dal suffisso `b`, oppure `k` per i kilobyte. Ad esempio, `find ~ -size -500b` cerca nella `/home` i file di dimensioni inferiori a 500 byte.

L'opzione `-S` consente di ordinare i file in base alle dimensioni, in ordine decrescente. Per trovare quelli di dimensione minori si adotta invece l'opzione `-Sr` (che inverte l'elenco). In questi casi l'opzione `-l` fa sì che sia possibile leggere anche le dimensioni dei file.

L'opzione `-empty` trova i file vuoti, e può essere utile per cancellarli.

Le opzioni `-mtime` e `-mmin` cercano i file modificati in un lasso di tempo indicato rispettivamente in ventiquattro ore e in minuti, posto come numero dopo l'opzione. Anche in questo caso si possono stabilire corrispondenze maggiori, uguali o minori degli argomenti forniti con una procedura del tutto analoga a quella indicata per `-size`. Ad esempio, `find /opt -mtime -1` cercherà quei file in `/opt` che sono stati modificati entro le ultime 24 ore, mentre `find /usr/bin -mmin 5` cercherà quei file in `/usr/bin` che sono stati modificati esattamente 5 minuti prima della ricerca.

L'opzione `-daystart` cerca i file modificati a partire dal giorno corrente. È inoltre possibile fissare come criterio di ricerca un intervallo; ad esempio, il comando `find ~ -mtime 2 -mtime 4 -daystart` cerca i file modificati da due a quattro giorni prima di quando viene effettuata la ricerca.

Un'opzione utilissima soprattutto negli script è `-exec`. Serve per eseguire il comando passato come argomento di `-exec` ai file trovati da `find`. Ad esempio, per trovare nella directory `/documenti` tutti i file di estensione `.txt` e per visualizzare le righe di questi file che contengono la stringa «parola» si darà il comando:

```
find ~/prova/ -name '*.txt' -exec grep parola '{} '\;
```

Se invece si vuole che `find` richieda la conferma dell'esecuzione per ciascun file trovato, in luogo dell'opzione `-exec` si adotta `-ok`. Per svolgere la stessa operazione, con richiesta di conferma per ogni visualizzazione, si darà il comando:

```
find ~/prova/ -name '*.txt' -ok grep parola '{} '\;
```

Vi sono ancora molte opzioni, per cercare i file secondo il proprietario, i gruppi, i permessi e altre ancora, per le quali si faccia ricorso alla pagina `man` di `find`.

Il comando locate

Il quarto è il comando `locate`. Esso produce i medesimi risultati di `find`, ma piuttosto che ricercare nel sistema opera mediante un database (presente in `/var/lib/slocate/slocate.db`) ed è perciò decisamente più veloce. Com'è ovvio, il database non esiste già, e va creato col comando `locate -u[v]`; l'opzione `-v` può essere utile per seguire lo sviluppo del processo, che è tanto più lungo quanto più numerosi sono i file presenti nel sistema e che registra tutti i file delle unità montate.

`locate` è configurato per essere aggiornato automaticamente alle 4:40AM, ma può essere aggiornato anche manualmente col comando `updatedb [-v]`.

Tale comando non richiede parametri aggiuntivi, solo il nome del file da cercare.

18.3.5 Creare un file nuovo

I file vanno ovviamente creati con le relative applicazioni. Tuttavia, il comando `touch` può essere utile per creare un file vuoto, atto a fare da segnaposto, ed eventualmente per essere editato in un secondo

momento. Se il file esiste già, `touch` aggiorna il *timestamp*, cioè la data e l'ora di creazione a quella corrente.

18.3.6 Eliminare un file

Il comando di base per cancellare un file è `rm`.

Il comando `rm -f` cancella forzatamente, cioè senza richiedere conferma, mentre `rm -i` cancella in maniera interattiva, cioè chiedendo la conferma per ogni file. L'opzione `-r` o `-R` consente la cancellazione ricorsiva di una directory e delle sue subdirectory. Di conseguenza `rm -rf *` cancellerà tutto il contenuto del disco fisso. È necessario prestare qui la massima attenzione, poiché un banale errore di digitazione può produrre conseguenze assai gravi. Così, `rm *.c` cancella tutti e solo i file di estensione `.c`, ma `rm * .c` cancella tutti i file poiché, a causa dello spazio fra `*` e `.c`, `.c` viene considerato come un secondo argomento (insignificante) di `*`.

18.3.7 Dividere e ricomporre un file

Il comando `split` serve a scomporre un file in più file di dimensioni minori, fra l'altro per poterlo trasferire in floppy o poterlo spedire come una serie di allegati a delle e-mail. Il comando accetta, fra le altre opzioni, le dimensioni in byte (`-b numero_byte`), kilobyte (`-k numero_kilobyte`) o megabyte (`-m numero_megabyte`) dei segmenti in cui dividerà il file. Inoltre, se non si precisa un file di destinazione, esso crea una serie di segmenti in cui nome è costituito di default dal prefisso `x` e il suffisso dalla sequenza `aa, ab, ..., an`. Se invece si precisa un nome di destinazione, i segmenti avranno quel nome più il suffisso `aa, ab, ..., an`. Ad esempio, il comando: `split -b 50k prova.txt` divide il file `prova.txt` in segmenti di 50 kilobyte chiamandoli rispettivamente `xaa, xab, ..., xbn`; invece, il comando `split -b 2m prova.txt prova` lo divide in segmenti di 2 MB chiamandoli rispettivamente `provaaa, provaab, ..., provaan`. Si può facilmente mantenere anche l'estensione originale del file: ad esempio, il comando: `split -b 100k testo.swx testo.swx.` produrrà una serie di segmenti chiamati rispettivamente `testo.swx.aa, testo.swx.ab, ..., testo.swx.an`.

Per ricomporre il file si usa il comando `cat`, con la sintassi: `cat file_origine > file_destinazione`. Ad esempio, `cat testo.swx.* > testo.swx` ricostruisce il file originale, dopo di che è possibile cancellare i file di origine

18.3.8 Spostarsi nelle directory

Il comando `cd` consente di spostarsi nelle directory. Accetta come argomento il nome della directory di destinazione. Il flag `.` (punto) specifica la directory corrente, il flag `..` (due punti) specifica la directory superiore a quella corrente. Di conseguenza, per spostarsi da una subdirectory a un'altra della stessa directory, il comando sarà `cd ../nuova_directory`. Senza specificazioni, `cd` sposta alla home dell'utente, e vale come `cd~`. Il comando `cd -` sposta all'ultima directory visitata (il cui nome viene visualizzato). Se necessario, il comando `pwd` (acronimo di «Print Working Directory») visualizza il nome della directory corrente.

18.3.9 Creare una directory

Il comando `mkdir` crea una nuova directory, fornendo come argomento il percorso, altrimenti essa viene creata a partire dalla corrente. Se si vuole creare una directory con subdirectory (cioè un albero di directory), basta scrivere il percorso completo. Se la directory di base esiste già, si deve aggiungere l'opzione `-p`. Se si vogliono creare contemporaneamente più subdirectory di una directory, esse vanno spaziate oppure specificate fra graffe e separate da virgole. Il comando risulta pertanto: `mkdir [-p] /dir/{subdir1,subdir2,...,subdirn}`

18.3.10 Cancellare una directory

Il comando `rmdir` cancella una directory solo se è vuota; se essa contiene dei file, il comando fallisce. In questo caso, si deve adoperare invece il comando `rm`, eventualmente accompagnato dall'opzione `-r` se la directory contiene delle subdirectory, per effettuare un'eliminazione ricorsiva. L'opzione `-f` rimuove tutti i file senza chiedere conferma, mentre l'opzione `-i` chiede conferma per ciascun file. Se, in modo improprio, sono specificate entrambe le opzioni `-f` e `-i`, il comando considera quella indicata per ultima.

18.4 L'uso dei caratteri jolly

La gestione dei file e delle directory può avvalersi anche dei cosiddetti «caratteri jolly»: i principali sono l'asterisco `*`, le parentesi quadre `[]` e l'apice semplice `'`. Si adottano anche altri operatori, come le graffe `{ }`.

Il simbolo `*` indica qualsiasi tipo e numero di caratteri arbitrari. Pertanto, `nome*` si riferisce a tutto ciò (file e/o directory) che inizia con la stringa «nome»; `tty*` indica tutti i file `tty`; `doc*` potrebbe indicare i file `doc1`, `doc2`, `documento1`, `documento2`, `doc_1`, `doc_2` ecc.; `m*d*` potrebbe indicare i file `miodocumento`, `martedì`, ecc.; `*.c` indica tutti i file di estensione `.c`. L'espressione `*.*`, tipica di Windows per indicare tutti i file, non va usata nello stesso modo, in quanto in GNU/Linux essa indica soltanto tutti i file e le directory contenenti un carattere punto (`.`).

Il simbolo `'` indica un solo carattere alfanumerico arbitrario. Dunque `nome'xx` indica tutti i file o directory che iniziano con la stringa «nome», terminano con la stringa «xx» e hanno nel mezzo uno e un solo qualsiasi carattere.

Il simbolo `[]` indica tutti i valori indicati o compresi tra graffe; se separati dal trattino, tali valori indicano l'intervallo compreso fra i due estremi. Dunque `nome[1-4]` indica tutto ciò che inizia con la stringa «nome» e che termina con un numero da 1 a 4 (perciò: «nome1», «nome2», «nome3», «nome4»). La stringa `nome[1a]` indica tutto ciò che inizia con la stringa «nome» e che termina col numero da 1 o con la lettera `a`. Ad esempio, in una directory contenente i file `doc1`, `doc2`, `documento1`, `documento2`, `docA`, `docB`, la stringa `doc[1A]` identificherebbe soltanto i file `doc1` e `docA`. Una lettera preceduta da punto esclamativo invece esclude quel carattere. Pertanto `a[!b]c` indica tutti i file che iniziano con `a` e terminano con `c` e che comprendono al loro interno ogni carattere, tranne `b`.

Le graffe inseriscono sulla parte precedente, su quella seguente oppure su entrambe. Ad esempio, il comando: `touch prova{1,2,3}` crea i file `prova1`, `prova2` e `prova3`, mentre il comando: `touch opera{tore,zione}.txt` crea i file `operatore.txt` e `operazione.txt`.

I caratteri jolly possono essere usati anche in combinazione: il comando: `ls *. [co]` avrà come output l'elenco di tutti i file la cui estensione è `.c` oppure `.o`.

Se si deve usare uno di questi caratteri come tale e non come jolly, cioè non si vuole che la shell lo espanda, lo si deve inserire fra virgolette semplici. Ad esempio, il comando: `echo *` stampa a video il contenuto della directory corrente, mentre se si vuole stampare proprio e solo l'asterisco, si darà il comando `echo '*'`. Se il testo contiene virgolette semplici da non espandere (come un apostrofo), la stringa andrà a sua volta racchiusa fra virgolette doppie. L'opzione `noglob`, infine, disattiva stabilmente i caratteri speciali della shell. Una volta impostata tale opzione, i caratteri `*`, `'`, `[]` e `~` non verranno più espansi se presenti all'interno di un file. Ciò serve qualora ci si voglia riferire a dei file che contengono, nel loro nome, tali caratteri.

18.5 Comandi intermedi

18.5.1 Cercare del testo

Per visualizzare, cioè stampare a video, delle righe tratte da un file o da qualche input che corrispondono a una determinata espressione, si usa il comando `grep`. Ad esempio, `grep root /etc/passwd` mostra tutte e solo le righe del file `/etc/passwd` che contengono la stringa «root». `grep` individua e comunica anche il nome dei file, se si lavora con file multipli. Invece il comando: `grep root /etc/*` fornirà tutte le righe dei file di `/etc` dove compare la stringa «root», stampando anche il nome dei file che la contengono; se si desidera vedere le righe di testo, ma non i nomi dei file, si deve aggiungere l'opzione `-h`. È possibile effettuare una ricerca ricorsiva specificando l'opzione `-r`. Fra le innumerevoli opzioni di questo comando, le più usate sono `-i`, per confronti non sensibili alle maiuscole e minuscole, e `-v`, che inverte la ricerca stampando tutte le righe che non contengono la stringa specificata. Se oggetto della ricerca non è una stringa, ma una parola intera, andrà aggiunta l'opzione `-w`. Se, ad esempio, si volesse cercare nella directory `/packages` (contenente dei pacchetti coi relativi SlackBuild) in quali SlackBuild è adottato `sed`, scrivere semplicemente `grep sed /packages/*/*.SlackBuild` restituirebbe tutte le righe in cui compare la stringa «sed», e dunque anche «Heavily based on Slackware 11.0 SlackBuild». Il comando corretto sarà allora `grep -w sed /packages/*/*.SlackBuild`.

Facciamo qualche esempio: per ricercare le righe di tutti i file della directory corrente contenenti la parola «word» il comando sarà: `grep word *`. Per ricercare le righe di tutti i file `.txt` della directory `~/testi` contenenti la parola «word», eliminando però i nomi dei file che contengono tali righe, il comando sarà: `grep -h word ~/testi/*.txt`. Per ricercare le righe di tutti i file `.txt` della directory `~/testi` e delle sue subdirectory contenenti la parola «word», il comando sarà: `grep -r word ~/testi/*.txt`.

Per ricercare una stringa di più parole, essa va racchiusa fra virgolette semplici.

Va tenuto presente che, in questo modo, `grep` cerca solo le stringhe contenute in una singola riga, mentre non trova quelle divise tra la fine di una riga e l'inizio della riga successiva. Se, ad esempio, io cercassi in un file la stringa «compact disc», ma «compact» fosse l'ultima parola di una riga e «disc» la prima della riga successiva, `grep` non la individuerrebbe.

18.5.2 Le espressioni regolari

Il comando `grep` è anche in grado di ricercare modelli di testo più complessi, denominati espressioni regolari (o *regexp*). Esse contengono dei metacaratteri che non rappresentano se stessi, ma vengono espansi per creare modelli dotati di più ampia validità e per consentire di conseguenza ricerche maggiormente elastiche e sofisticate nonché manipolazioni di stringhe di testo. Sintatticamente, per indicare che i simboli di cui le espressioni regolari fanno uso non sono caratteri, ma metacaratteri, essi vanno fatti precedere dal simbolo della barra (/).

Forniamo l'elenco dei metacaratteri:

- . individua la corrispondenza di un qualsiasi carattere non definito, a eccezione del carattere di nuova riga.
- * individua la corrispondenza con zero o più volte dell'espressione regolare (ad esempio, `-*` individua le righe di testo che contengono uno o più «-», indipendentemente dal loro numero).
- [] individua qualunque elemento dell'insieme o intervallo di caratteri (ad esempio, `[abc]` trova la corrispondenza con `a`, `b` o `c`. Il trattino specifica un intervallo, per cui `[1-4]` individua 1, 2, 3 e 4. L'accento circonflesso trova un intervallo tranne la stringa posta dopo `^` per cui `^[a]` troverà tutte le lettere, tranne «a» e `^0-9` qualunque carattere, tranne le cifre).
- ^ individua la corrispondenza con l'inizio di riga (ad esempio, `^a` individua «a» solo quando essa è il primo carattere della riga).

`$` individua la corrispondenza con la fine di riga (ad esempio, `$a` individua «a» solo quando esso è l'ultimo carattere della riga).

`\` individua il metacarattere come carattere letterale.

L'uso di queste espressioni consente di effettuare molteplici ricerche. Abbiamo visto che per visualizzare le righe che iniziano con un determinato testo si utilizza `^`. Ad esempio, il comando: `grep ^abc /home/documenti/file.txt` cercherà tutte le righe di `file.txt` che iniziano con la stringa esatta «abc». Invece il comando: `grep -i ^abc /home/documenti/file.txt` cercherà tutte le righe di `file.txt` che iniziano con la stringa «abc» indipendentemente dalle maiuscole o minuscole.

Per visualizzare le righe che finiscono con un determinato testo si utilizza `$`. Ad esempio, il comando: `grep $abc /home/documenti/file.txt` cercherà tutte le righe di `file.txt` che terminano con la stringa esatta «abc». Invece il comando: `grep -i $abc /home/documenti/file.txt` cercherà tutte le righe di `file.txt` che terminano con la stringa «abc» indipendentemente dalle maiuscole o minuscole (grazie all'opzione `-i` di `grep`).

Per visualizzare le righe di una determinata lunghezza si usa il punto (`.`) racchiuso fra i simboli dell'accento circonflesso (`^`) e del dollaro (`$`); tanti devono essere i punti quanti i caratteri. Ad esempio, il comando: `grep ^...$ /home/documenti/file.txt` cercherà tutte le righe di `file.txt` che hanno una lunghezza di tre caratteri. Ciò è naturalmente scomodo se le righe sono composte da numerosi caratteri. In questo caso conviene usare il costrutto `^.numero_righe$`, mentre il simbolo della virgola (`,`) specifica un intervallo. Così, mentre `grep ^.\25$ /home/documenti/file.txt` cercherà in quel file le righe lunghe 25 caratteri e infine `grep ^. \ 25,\ \ /home/documenti/file.txt` cercherà le righe pari a venticinque o più caratteri.

Per visualizzare le righe che contengono una qualsiasi espressione regolare, si usa `|`, ossia l'operatore di alternanza. Ad esempio, il comando: `grep acb|def /home/documenti/file.txt` cercherà le righe di `file.txt` dove compaiono le stringhe «abc» o «def».

18.5.3 Vedere del testo

Per leggere convenientemente un testo lungo, che scorre velocemente sullo schermo, si usano `more` e `less`. Entrambi i comandi possono apparire come ultimo comando di una pipeline, per poter esaminare l'output di una serie di comandi.

Il comando `more` fa apparire una pagina di testo alla volta. Premendo **Invio** si avanza di una linea, e premendo la barra spaziatrice si passa alla schermata successiva. Per uscire da `more` si usa **q**.

Il comando `less` svolge le medesime funzioni, ma è più potente di `more`, in quanto consente anche di ritornare indietro premendo il tasto **b**. Per uscire da `less` si usa **q**. `Less` consente anche di ricercare delle stringhe all'interno del testo: per ricercare una stringa in avanti si digita `/stringa` e si dà **Invio**, mentre per ricercare una stringa indietro si digita `?stringa` e si dà **Invio**. Tutte le occorrenze verranno evidenziate. `Less` è in grado anche di aprire più file contemporaneamente, che verranno visualizzati in successione; per spostarsi al successivo si digita `:n`, per spostarsi al successivo si digita `:p`.

Infine, nella forma `zless`, consente di leggere file di testo compressi con `gzip` senza doverli prima decomprimere. È possibile anche utilizzare `cat`, il quale però presenta una certa scomodità nel caso di file lunghi perché non ne potrà far vedere che la parte finale, a meno di non redirigerlo a sua volta a `more` o `less`, oppure ci obbliga a risalire col cursore o con la rotellina del mouse.

18.5.4 Vedere una porzione di testo

Per leggere una porzione di un testo si usano `head`, `tail`, e `sed`, che stampano rispettivamente le prime dieci righe (`head`), le ultime dieci righe (`tail`), e una o più righe a piacere (`sed`). Con l'opzione `-n`, dove `n` è un numero a piacere, `head` e `tail` visualizzano invece `n` righe. L'opzione `+n` di `tail` fa invece

iniziare la visualizzazione dalla riga *n*. È anche possibile vedere la parte finale di un file in progressione, cioè scritto da un altro processo, e il cui output si modifica continuamente; in questo caso il processo si interrompe col comando **Ctrl+c**. Il comando `echo`, invece, visualizza un testo specificato a schermo; la stringa da visualizzare va specificata dopo il comando stesso. Di default, `echo` visualizza la stringa e stampa una nuova linea vuota dopo la stringa, a meno che non si aggiunga l'opzione `-n`, la quale impedisce alla riga di essere inviata allo standard output.

18.5.5 Ordinare delle righe di testo

Per ordinare delle righe di un file in ordine alfanumerico si usa il comando `sort`. Se le righe iniziano con dei numeri e si vogliono mettere anche questi in ordine, si adotta l'opzione `-n`. Per ordinare in ordine inverso, si usa l'opzione `-r`. L'opzione `-u` serve a eliminare le righe duplicate, mostrando soltanto la prima che coincide col testo. Infine, l'opzione `-f` consente ricerche *case insensitive*, ossia tali da ignorare la differenza fra maiuscole e minuscole.

18.5.6 Contare il testo

Il comando `wc` conta il testo, indicandone nell'ordine le righe, le parole e i caratteri. Con l'opzione `-l` conta solo le righe, con l'opzione `-w` conta solo le parole, con l'opzione `-c` conta solo i caratteri.

18.5.7 Concatenare del testo

Il comando `cat` unisce più file di testo o file binari. Serve perciò per fondere due o più file in un nuovo file; pertanto il comando `cat file1 file2 > file3` creerà il nuovo `file3` contenente in sequenza il testo di `file1` e di `file2`. Il comando `cat file2 file1 > file3` avrebbe invece creato un file `file3` contenente in sequenza il testo di `file2` e quindi di `file1`. Questo comando serve anche per creare un nuovo file di testo in maniera veloce, senza aprire un editor, reindirizzando il suo output al file specificato con l'operatore `>`. Si diano i comandi che seguono:

```
cat > file1 [Invio]
aaa aaa aaa [Invio]
bbb bbb bbb [Invio]
Ctrl+d
```

In tal modo si crea il file `file1` che conterrà due righe di testo (quelle indicate nella seconda e terza riga del comando); per chiudere l'operazione si digita (come nei comandi tipici di Emacs), il comando **Ctrl+d**. Gli inconvenienti stanno nel fatto che, se si commette un errore, le uniche via di uscita consistono nel comando **Ctrl+u** per cancellare l'intera riga o addirittura **Ctrl+c** per abbandonare il file senza scrivere nulla, ma tutto il resto del lavoro andrà perduto.

Infine, il comando `cat` serve per aggiungere del testo alla fine di un file, reindirizzando il suo output con l'operatore `>>` (se venisse usato invece l'operatore `>` il file destinazione verrebbe sovrascritto). Ad esempio, per aggiungere la riga «ccc» alla fine di `file1`, contenente le righe «aaa» e «bbb», si daranno i comandi:

```
cat >> file1 [Invio]
ccc [Invio]
Ctrl+d
```

18.5.8 Trovare il formato di un file

Dal momento che in GNU/Linux non si usano necessariamente le estensioni per caratterizzare i tipi di file, a volte è necessario cercare di individuarne il formato. A ciò serve il comando `file nome_file`,

che spesso riesce in effetti a individuare correttamente di quale tipo di file si tratta. Ad esempio, il comando: `file /boot/boot_message.txt` darà come output:

```
bash-3.00$ file boot_message.txt
boot_message.txt: ASCII English text
```

Se il file è compresso con `gzip`, per scoprire il formato del file originale si adotta l'opzione `-z`. Ad es., l'output di `file` per un file di questo tipo sarà:

```
samiel@darkstar:~$ file -z slack4dummies.tex.gz slack4dummies.tex.gz: LaTeX
2e document text (gzip compressed data, was "slack4dummies.tex", from Unix,
max compression)
```

18.5.9 Registrare una sessione di shell

Il comando `script` registra una sessione di shell in un file. Il comando sarà del tipo `script nome_file;` nel file in questione verranno riportati i comandi digitati e il relativo output, nonché data e ora dell'inizio e della fine della registrazione. In questo modo si può tenere stabilmente traccia delle procedure effettuate. Per uscire dalla registrazione si digita `exit nome_file`.

18.5.10 Ottenere informazioni sull'hardware

Il comando `lspci`

Il comando di base per ottenere informazioni sull'hardware della macchina è `lspci`, che fornisce un riepilogo. Informazioni progressivamente più dettagliate si ottengono con le opzioni `-v` e `-vv`. In alcuni casi, le periferiche molto vecchie non sono identificate da `lspci`, e allora delle informazioni possono essere fornite comunque da `dmesg`.

Il comando `dmesg`

Se quanto cerchiamo non sono solo informazioni sulle periferiche PCI, ma su tutto il sistema (dunque relative anche a periferiche USB, SCSI, configurazione della memoria, CPU ecc.), allora ricorreremo al comando `dmesg`, che registra tutto quello che viene rilevato dal kernel. Come abbiamo visto, la lettura dell'output viene facilitata reindirizzandolo mediante `| less`. È inoltre possibile filtrare l'output con `grep`, se si intendono cercare informazioni su un solo tipo di periferiche. Si avranno così, ad esempio, i comandi:

```
dmesg | grep -i usb
dmesg | grep -i isa
dmesg | grep -i hd
dmesg | grep -i memory
dmesg | grep -i cpu
e via dicendo.
```

Conoscere lo stato dell'hardware con `/proc`

Per seguire il sistema in esecuzione e monitorare alcuni elementi come la memoria fisica, l'uso della CPU e altro si legge il file system virtuale `/proc` con `cat`. Si può anche esplorare `/proc` come una directory qualsiasi col comando `ls`, e quindi ottenere le informazioni con una serie di comandi come:

```
cat /proc/cpuinfo
cat /proc/meminfo
cat /proc/modules
cat /proc/partitions
cat /proc/scsi/scsi
```

e via dicendo. Per assumere tali informazioni, bisogna affidarsi per l'appunto a `cat`, oppure a utilità progettate espressamente come `sysctl`, `lspci`, `ps` e `top`; pager come `less` e `more` producono un quadro lievemente diverso, perché rileggono `/proc` a ogni pagina. Ricorrere invece a un editor di testo o a un'utilità capace di scrivere su disco è molto pericoloso, perché potrebbe compromettere l'intero sistema. Una curiosità: se si vuol sentire il rumore prodotto dalla CPU, si può lanciare il comando `cat /proc/kcore > /dev/dsp`, magari con gli altoparlanti a volume basso...

18.5.11 Ottenere informazioni sul sistema

Il comando `du` mostra lo spazio occupato da un file o da una directory specificate in blocchi da 512 byte oppure in blocchi da 1024 byte se si aggiunge l'opzione `-k`; senza la precisazione di file o directory, mostra i dati relativi alla directory corrente.

Il comando `fdisk -l` mostra tutte le partizioni del disco o dei dischi fissi. Se si precisa una partizione, mostra esclusivamente le informazioni relative a quella.

Infine, il comando `free` fornisce diverse informazioni sulla memoria: quella totale, quella usata, quella libera, e sull'uso o meno del file di swap.

18.6 Redirezione dell'input e dell'output

18.6.1 Redirezione dello standard output (stdout)

La shell consente la redirezione dell'output, che come sappiamo è il terminale stesso (stdout). Per ottenerlo, si usa il carattere di redirezionamento `>`. Ad esempio, per redirigere l'output verso un file, si usa *comando* `> file`. Se voglio che l'elenco dei file di una directory non venga stampato a video, ma nel file `elenco.txt`, posso scrivere `ls > elenco.txt`. Il file viene così creato. Se esiste già un file con tale nome, esso viene sovrascritto. Per evitare la sovrascrittura è possibile impostare la funzionalità `noclobber` col comando `set -o noclobber`, che, in caso il file esista già, dà un messaggio di errore. Anche se è attivo `noclobber`, è comunque sempre possibile sovrascrivere il file vecchio aggiungendo, dopo il carattere di redirezionamento, un punto esclamativo, ad esempio `ls >! elenco.txt`. Un'altra possibilità è usare il carattere di redirezionamento `>>`, con la sintassi *comando* `>> file`. In questo modo il nuovo stdout verrà aggiunto all'interno del vecchio file, alla fine del testo preesistente.

Bisogna solo prestare attenzione al fatto che l'operazione di redirezione viene effettuata prima del comando. Seguiamo questi passaggi:

```
bash-3.1$ ls
file1 file2 file3
bash-3.1$ ls > elenco
bash-3.1$ cat elenco
elenco
file1
file2
file3
```

Dunque in `ls > elenco` il file `elenco` è stato creato prima che venisse eseguito il comando `ls`. Allo stesso modo, se fosse presente un file `elenco.txt` con dei dati, e venisse impartito il comando `cat elenco.txt >> elenco.txt`, usando dunque lo stesso file sia come input sia come output, `cat` mostrerebbe

un file vuoto perché ha innanzitutto creato un nuovo file `elenco.txt` (vuoto) sovrascrivendo il vecchio, e quindi ne ha mostrato il contenuto (nullo).

18.6.2 Redirezione dello standard input (stdin)

È anche possibile redirigere lo standard input, benché sia piuttosto raro dal momento che gran parte dei comandi di GNU/Linux accetta i nomi di file come argomenti. Ad esempio, il comando `head /proc/cpuinfo` potrebbe essere riscritto come: `head < /proc/cpuinfo`. Lo `stdin` può essere rediretto per ottenere i dati da un file, anziché dalla tastiera; ad esempio, il comando `cat < file.txt` mostra il contenuto del file in questione.

18.6.3 Redirezione dell'errore (stderr)

È spesso comodo redirigere a un file (o magari alla stampante) l'errore standard (`stderr`), un flusso aggiuntivo a scopo diagnostico. Proviamo a dare il comando `ls /aaaaaaa`; poiché nella nostra ipotesi questa directory non esiste, a schermo leggeremo: «No such file or directory». Se redirigiamo lo `stdout` a un file, col comando: `ls /aaaaaaa > elenco.txt`, troveremo questo file ovviamente vuoto, ma continueremo a leggere sullo schermo il messaggio di errore. Per redirigere quest'ultimo a un file, si adotta la formula `2>`. Il comando diventa allora: `ls /aaaaaaa > elenco.txt 2> errori.txt`. Infatti `>` indica lo stream `ID1`, che è lo `stdout`, mentre `2>` indica lo stream `ID2`, e cioè l'errore standard. A questo punto avremo creato il file `elenco.txt` sempre vuoto, e il file `errore.txt` contenente il messaggio di errore che prima compariva a schermo. Anche in questo caso l'operatore `2>>` aggiunge l'output al file preesistente, anziché sovrascriverlo.

È infine possibile redirigere più flussi allo stesso file, ad esempio sia lo `stdout` sia lo `stderr` col carattere di redirezione `&>`; avremo così, ad esempio, il comando: `ls /aaaaaaa > elenco.txt 2>&1`.

Ciò può essere utile, come abbiamo visto, quando si compila un pacchetto di ampie dimensioni e non si vuole stare sempre davanti al video per controllare eventuali messaggi che, inoltre, scorrono assai veloci. Alla fine, leggeremo con comodo o l'intero processo (ad esempio, `sh nano.SlackBuild --cleanup > nano.txt`) o soltanto gli eventuali errori (`sh nano.SlackBuild --cleanup 2> errori_nano.txt`)

18.6.4 Il piping

Per redirezionare il comando non a un file, bensì a un altro comando, si usa la pipe, cioè l'operatore `|`. Si digiti il comando `head /proc/cpuinfo`. L'output consiste in alcune informazioni sulla CPU del computer, che leggiamo a schermo. Se invece dalla nostra home diamo il comando `head /proc/cpuinfo > cpuinfo.txt`, troveremo un file con quel contenuto, compiendo una semplice redirezione. Ma se digitiamo `head /proc/cpuinfo | tr a-z A-Z` (il comando `tr` cambia le maiuscole in minuscole), e cioè redirigiamo l'output di `head` al comando `tr`, otterremo a schermo che tutte le lettere minuscole (a-z) sono state trasformate in maiuscole.

Allo stesso modo, il comando `ls | lpr` redirige l'output del comando `ls` alla stampante invece che allo schermo (e cioè stampa l'elenco invece di visualizzarlo), mentre per stampare il testo di un file si darà `cat file.txt | lpr`. Si possono infine combinare anche più comandi. Ad esempio, `sort elenco.txt | cat -n | lpr` legge il file `elenco.txt` e lo ordina, quindi lo invia alla stampante.

18.7 Ottenere la guida

18.7.1 Le pagine man

La guida di base dei comandi GNU/Linux è costituita dalle pagine man, a cui si accede col comando `man comando`. Le opzioni sono indicate sistematicamente, senza mettere in rilievo quelle fondamentali e quelle invece accessorie. Per cercare una pagina man in base a una parola chiave, si usa l'opzione `-k`, che offre l'elenco di tutte le pagine rilevanti. Le pagine man sono divise in sezioni numerate, secondo un criterio generale, per cui quando ci si riferisce a una determinata pagina fra parentesi compare anche il numero di sezione:

Sezione	Descrizione
1	Comandi utente
2	Chiamate di sistema a basso livello
3	Documentazione sulla libreria di programmazione Unix di livello superiore
4	Informazioni sulle interfacce e i driver di periferica
5	Descrizione dei file di configurazione del sistema
6	Giochi
7	Formati di file, convenzioni e codifiche (ASCII, suffissi ecc.)
8	Comandi di sistema e server

Se un comando viene considerato in più sezioni, si può leggere quella che interessa precisandola fra `man` e *nome_comando*, ad esempio: `man 2 mount`.

Le pagine man possono essere visualizzate col supporto del colore ricorrendo a un pager (cioè un'utilità di impaginazione) come `most`. In `/etc/profile` andrà inserita la riga:

```
export PAGER="most"
```

e di conseguenza i nomi dei file e i link ad altre pagine man appariranno di un colore diverso rispetto a quello del testo, facilitando la consultazione, mentre si vedrà il numero della riga in cui è posizionato il cursore e la percentuale del testo già scorso.

18.7.2 Stampare una pagina man

Per stampare in forma accettabile una pagina man, si aggiunge l'opzione `-t` al fine di migliorarne l'output e lo si reindirizza tramite una pipe verso la stampante PostScript. Ad esempio, il comando: `man [-t] bash | lpr` stampa la pagina man di `bash` e, con l'opzione `-p`, formatta il risultato per migliorarne la leggibilità.

18.7.3 Il comando `whatis`

Il comando `whatis` restituisce una sintesi (di una riga) della pagina man. È inoltre possibile ottenere una guida digitando il nome di un comando seguito dall'opzione `--help` o in forma abbreviata `-h` (comandi diversi supportano l'una o l'altra).

18.7.4 Il sistema di documentazione `info`

In GNU/Linux vi è inoltre un più recente sistema di documentazione, in formato `texinfo` (prodotto con `makeinfo` e compresso con `gzip`, ma che viene decompresso al volo per la lettura), più ampia delle

pagine `man`, tale che spesso corrisponde a un manuale vero e proprio. Per sapere quali testi `info` sono presenti nel sistema e quali sono i comandi principali, si digita `info` senza opzioni. Per leggere un manuale `info`, si digita `info comando`. In questo caso, se la pagina esiste, si apre il cosiddetto *nodo top*, che presenta i collegamenti ai vari capitoli del manuale o *nodi info*. Ogni nodo `info` contiene i collegamenti ai paragrafi, e così via. I collegamenti sono indicati dal simbolo dell'asterisco (*), seguito dal nome del nodo e da uno o due simboli del doppio punto (: :). Per andare a un nodo `info` si usano le frecce e quindi **Invio**, mentre per uscire si usa il tasto **q**. Per leggere un file specifico si adotta l'opzione `-f`, e per leggere un nodo specifico l'opzione `-n`. Una pagina `info` non presente nella posizione canonica, e cioè in `/usr/info`, può comunque essere decompressa al volo e letta aggiungendo l'opzione `-r`. Se invece la pagina non esiste, `info` presenta la corrispondente pagina `man`.

18.7.5 Ricerche con apropos

Per cercare aiuto su argomenti non ben delimitati o di cui abbiamo solo dei termini di riferimento, ci si può valere del comando `apropos`, che funziona in modo analogo a `man -k`, restituendo un elenco di comandi relativi a un argomento. Questo strumento è particolarmente utile quando si cerca un'applicazione di cui non si ricorda il nome.

18.8 Personalizzare la shell bash

18.8.1 Le impostazioni generali di bash

La shell `bash` presenta delle impostazioni standard generali per tutti gli utenti, ma ogni utente può personalizzare la propria shell modificando opportunamente alcuni file di configurazione. In questi ultimi sono presenti delle variabili a cui vengono assegnati dei valori predefiniti, creati al momento del login ed eliminati al momento del logout. Ad esempio, la variabile `USER` mostra il nome dell'utente che ha effettuato il login.

Quando un utente accede al sistema, viene creata una sua shell di login che inizializza alcune variabili, secondo i valori predefiniti. L'utente può però creare delle variabili personali e modificare alcune di quelle predefinite (fra l'altro modificando il prompt, attribuendo colori particolari alla visualizzazione o creando degli alias per i comandi). Per convenzione, le variabili della shell sono definite con nomi costituiti da lettere maiuscole. Per visualizzare il contenuto di una variabile ci si vale del comando `echo` seguito dal carattere `$` e dal nome della variabile. Ad esempio, un determinato utente può visualizzare la variabile `HOME` utilizzando il comando: `echo $HOME`, che avrà come output il percorso della sua home. Allo stesso modo è possibile impostare una shell di logout, che effettua determinate operazioni (come cancellare i file temporanei) all'uscita dal sistema.

Scheda n. 27

Le shell di Slackware

In Slackware sono presenti le seguenti shell:

- **bash** la shell Bourne Again
- **tcsh** una shell compatibile con `csh`
- **csh** la shell C, un collegamento simbolico a `tcsh`
- **ash** una piccola shell di tipo `sh`

- **ksh** la shell Korn
- **zsh** una shell compatibile con csh, ksh e sh

Le C shell (csh) sono un tipo di shell usate nei sistemi Unix, originariamente sviluppate da Bill Joy per il sistema BSD. La shell C deriva dalla sesta versione di sh, che si è sviluppata quindi come shell Bourne, ma la sua sintassi è modellata su quella del linguaggio di programmazione C. La shell C è stata potenziata rispetto ai suoi predecessori, fra l'altro abilitando gli alias e la history dei comandi. Le shell C sono state sopravanzate da shell come la Tenex C shell (tcs), le Korn shell (ksh) e la bash. Le shell Bourne shell sono un tipo di shell utilizzato tipicamente sui sistemi Unix e Unix-like, come alternativa alle C shell. Il loro nome deriva dal loro inventore, Stephen Bourne; per convenzione, tutte le shell di tipo Bourne vengono indicate con il suffisso sh. La più celebre Bourne shell è bash (acronimo di «Bourne Again SHell»), sviluppata dalla Free Software Foundation col fine di riunire i vantaggi delle Bourne shell, C shell (csh) e Korn shell (ksh). Pur introducendo molte innovazioni nell'interazione con l'utente, lo scripting delle C shell è stato sottoposto a diverse critiche da più parti; poiché tuttavia la presenza di una Bourne shell è garantita in tutti i sistemi compatibili Unix, ha preso sempre più piede lo scripting di tipo sh.

18.8.2 Tipi di shell

In GNU/Linux sono presenti tre tipi di shell: la **shell di login**, le **shell interattive di non login**, le **shell non interattive** (di non login).

La **shell di login** cerca le proprie impostazioni e variabili nel momento in cui l'utente accede al sistema (quindi in Slackware di default al runlevel 3 e non nel server X). Essa legge il file `/etc/profile` e, se esistono, cerca nell'ordine `~/.bash_profile`, `~/.bash_login` e infine `~/.profile`. In realtà, in Slackware è possibile e anzi opportuno usare soltanto il primo (altre distribuzioni contengono già il file `.profile`, che equivale in Slackware al file `~/.bash_profile`). L'opzione `--noprofile` inibisce tale comportamento, cosicché la shell non legge questi file di configurazione. Quando si esce da questa shell, essa cerca di leggere il file `~/.bash_logout`, se presente. In questo file si specificano le operazioni che l'utente vuole eseguire alla chiusura della sessione.

Per quanto attiene invece alle **shell interattive di non login**, nel server X esse (a meno che non si precisi l'opzione `--norc`) leggono il file `~/.bashrc` (ad esempio quando si lancia `xterm` da un gestore di finestre come Konsole), e non `~/.bash_profile`. In tale file andranno specificati gli alias e alcune funzioni definite dall'utente. Se invece si decidesse di impostare la configurazione di queste shell in `~/.bash_profile`, si dovrebbe inserire in `~/.bashrc` un comando per leggere `~/.bash_profile`, ma questo scenario è piuttosto anomalo.

Per quanto attiene alle **shell non interattive di non login** di ogni specifico utente, per esse va creato il file `~/.bash_profile`. Le opzioni che quest'ultimo contiene prevalgono su quelle indicate in `/etc/profile`. Questo significa che eventuali modifiche a `/etc/profile` si ripercuoteranno su tutte le shell degli utenti, mentre le modifiche apportate a `~/.bash_profile` varranno soltanto per l'utente nella cui home è presente questo file. In tale file andranno specificate le variabili di ambiente (`BASH_ENV`).

I criteri di lettura dei file di configurazione spiegano anche la differenza fra il comando `su` (acronimo di «switch user») e il comando `su -`. Infatti, `su` avvia una shell interattiva di non login, per cui legge solo, se presente, `~/.bashrc`; le variabili di ambiente rimangono pressoché immutate rispetto a quelle dell'utente dalla cui shell è stato lanciato il comando; la conseguenza immediata è che ci ritrova root nella stessa directory di partenza. Lo si può vedere considerando, ad esempio, il fatto che, dando `su` ed elencate le variabili col comando `env`, la voce `USER` continua a dare il nome dell'utente, mentre dal comando `whoami` risulta root. Per contro, `su -` avvia una shell di login, per cui legge i tutti i file di configurazione presenti (e cioè `/etc/profile`, `~/.profile`, `~/.bash_profile`, `~/.bash_login` e – allorché si effettua il

logout – `~/bash_logout`; di conseguenza vengono reimpostate a root tutte le variabili di ambiente. Infatti a seguito del comando `su -` vengono eseguiti diversi compiti amministrativi, come impostare lo UID e il GID della tty. Se esiste, la variabile d'ambiente `TERM` viene conservata, a differenza delle altre variabili d'ambiente (a meno che non si specifichi l'opzione `-p`). Quindi vengono reimpostate le variabili d'ambiente `HOME`, `PATH`, `SHELL`, `TERM`, `MAIL` e `LOGNAME`. `PATH` viene predefinito come `/usr/local/bin:/bin:/usr/bin:` per gli utenti normali e come `/sbin:/bin:/usr/sbin:/usr/bin:` per root.

Come abbiamo accennato, vi sono infine le **shell di logout**, atte a compiere una serie di operazioni all'uscita dalla sessione. Può esserci una shell globale di logout, quella richiamata in `/etc/rc.d/r.0` alle righe:

```
# Run any local shutdown scripts:
if [ -x /etc/rc.d/rc.local_shutdown ]; then
    /etc/rc.d/rc.local_shutdown stop
fi
```

I comandi inseriti in questo file – e cioè `/etc/rc.d/rc.bash_logout`, da rendere eseguibile – vengono eseguiti comunque all'uscita di ogni sessione, indipendentemente dall'utente. Possono poi esserci shell personali di logout, inserite nella home dei vari utenti come `~/bash_logout`, che conterranno specifici comandi, ad esempio la cancellazione di file temporanei o altro.

18.8.3 Personalizzazione di Bash

Il file `/etc/profile` contiene una serie di impostazioni generali: imposta i `PATH`, la shell di default, la forma del prompt, l'`umask`, i colori della shell. Slackware presenta solo questo file, che riguarda le shell interattive di login di tutti gli utenti (e per questo non è situato in nessuna particolare home), lasciando all'utente il compito di creare gli altri. Il file `~/bash_profile` dovrebbe contenere la personalizzazione delle variabili e dei `PATH` e il file `~/bashrc` gli alias.

Se la configurazione delle shell di tutti gli utenti non necessita di alcuna distinzione fra shell di login e shell interattive di non login e non ci sono specifici parametri di login da impostare, allora basta far sì che tutte le shell leggano `~/bashrc` inserendo in `/etc/profile` il comando:

```
if [ -r ~/.bashrc ]; then
source ~/.bashrc
fi
```

Da un punto di vista sintattico, il costrutto *if...then* non è strettamente necessario, perché basterebbe passare il comando `source` senza alcuna condizione. In questo caso, se il file è presente verrà letto, se non è presente non verrà letto, ma non si riceverà nemmeno un messaggio di errore che potrebbe disturbare; se infine il file fosse presente, ma l'utente non avesse il permesso di leggerlo, allora e solo allora si riceverebbe un messaggio di errore, che però in questa situazione potrebbe essere significativo. In questo caso, il comando da inserire in `~/bash_profile` sarà:

```
echo "source ~/.bashrc" >> ~/.bash_profile
```

Se la configurazione della shell di un singolo utente non necessita di alcuna distinzione fra shell di login e shell interattive di non login, allora la soluzione più comoda è creare un link simbolico da `~/bashrc` a `~/bash_profile` (col comando: `ln -s ~/.bashrc ~/.bash_profile`), che risulterà pertanto un doppio del primo file. Così `~/bashrc` sarà letto comunque.

In entrambi questi casi però si rinuncia a effettuare una possibile differenziazione fra shell di login e shell interattive di non login, che invece potrebbero (e per certi versi dovrebbero) essere distinte.

Come alternativa, è possibile creare un `~/bash_profile` che contiene un richiamo a `~/bashrc`, contenente una configurazione generica, e quindi aggiungervi dei comandi specifici per il login. Infatti, in

questo `~/.bash_profile` dovrebbero essere presenti almeno le variabili di ambiente `ENV` e `BASH_ENV`, e inoltre i comandi relativi alla gestione della shell stessa che ricorrono a `stty`, eventuali modifiche o aggiunte al `PATH` (se invece compiute all'interno di `~/.bashrc`, queste ultime dovrebbero apparire come delle assegnazioni del tipo `"PATH=~/.bin:/usr/bin:..."` piuttosto che come la semplice definizione del `PATH` e dunque `"PATH=~/.bin:\$PATH..."`), la riassegnazione delle variabili speciali contenute in `/etc/profile` e l'assegnazione di variabili speciali nuove.

Alcune applicazioni fanno partire il terminale sempre e comunque come subshell o shell di login; ad esempio Konsole di KDE segue questo comportamento. In genere, è possibile distinguere empiricamente i due tipi di shell dal prompt: un prompt come `user@host:~/ $` indica che si tratta di una shell di login, mentre un prompt come `bash-3.00$` indica che si tratta di una shell di non login. Ciò si può verificare facilmente: personalizziamo in un certo modo il prompt della shell di login mediante `~/.bash_profile` e in un altro modo quello della shell interattiva di non login mediante `~/.bashrc`. Quindi lanciamo Konsole. Vedremo che il suo prompt è quello definito in `~/.bash_profile`. Se, sempre da Konsole, apriamo una sessione con la voce «Nuova console Linux» il prompt sarà quello definito in `~/.bashrc`. Si può aprire direttamente una shell di non login con `xterm` (presente in Slackware) oppure con `Aterm` o `Eterm` (che invece non vi sono contenute).

18.8.4 Gli alias dei comandi

L'alias non è che l'abbreviazione di un comando, eventualmente corredato di opzioni e argomenti. Invece di digitare `shutdown -h now`, io posso decidere di denominare questo comando «spegni». Digitando il comando `spegni` non faccio altro che dare il comando originario in forma abbreviata. Naturalmente l'alias supporta tutte le opzioni dei comandi. L'alias si crea da shell in forma volatile con la sintassi: `alias nome_alias='comando'`, ma in questo caso esso vale solo nella sessione presente di shell e va quindi perduto. Se invece viene dichiarato in un file di configurazione, la sintassi cambia in: `alias nome_alias="comando"` (gli apici sono cioè semplici singoli nella shell, doppi nel file di configurazione).

Per rimuovere un alias, è sufficiente dare il comando: `unalias nome_alias`.

Abbiamo visto l'ordine in cui la shell cerca i suoi file di configurazione. Di conseguenza, gli alias possono essere inseriti in tutti e tre i suddetti file, a seconda del risultato che si vuole ottenere.

Gli alias utili a tutti gli utenti (detti a volte *general purpose*) devono essere collocati in `/etc/profile` e saranno disponibili in modalità testuale.

Gli alias utili sia in modalità testuale sia in modalità grafica (cioè nella shell invocate dall'interno del server X) vanno inseriti in entrambi i file `~/.bashrc` e `~/.bash_profile`.

Gli alias *general purpose* utili anche in modalità grafica devono essere collocati non solo in `/etc/profile`, ma anche nel file `.bashrc` di ogni utente. Una soluzione efficiente consiste nell'inserire all'interno della directory `/etc/skel` un `.bashrc` e un `.bash_profile` valido per tutti gli utenti. Infatti la directory `/etc/skel` è assunta come modello (o *master*) per creare la cartella di default di ogni nuovo utente, e di conseguenza inserendo i due file appena citati in `/etc/skel`, si avrà che ogni nuovo utente creato avrà di default gli stessi alias.

Nelle shell non interattive gli alias non funzionano: pertanto negli script non è possibile utilizzarli, a meno di non trasformare la shell in interattiva con l'opzione `-i` (una shell interattiva si lancia col comando: `bash -i`): in questo caso gli script recheranno, alla prima riga, l'intestazione:

```
#!/bin/bash -i
```

Va infine ricordato che gli alias non accettano parametri in forma complessa. Per superare questo limite, bisogna ricorrere a delle funzioni. Ad esempio:

```
function gzless()  
{  
  gzip -cd \"$1 | less ;  
}
```

per cui poi sarà sufficiente impartire il comando `gzless nome_file`.

18.9 Le variabili

18.9.1 Tipi di variabili

Le variabili appartengono a tre gruppi principali:

- variabili di ambiente:** fanno parte del sistema e non serve definirle. In Slackware numerose variabili di ambiente sono definite in `/etc/profile`. Alcune di esse, come i path, possono essere modificate dall'interno di un programma di shell.
- variabili incorporate:** sono fornite dal sistema e non possono essere modificate
- variabili di utente:** sono definite dall'utente e possono essere modificate a piacere. In Slackware dovrebbero essere definite in `~/.bash_profile`.

Inoltre possiamo annoverare le variabili locali, che si distinguono variabili di utente o, in un senso ancora più ristretto, proprie di una sola sessione di shell.

18.9.2 Nome e valore delle variabili

Una variabile regolare viene definita all'interno di una sessione la prima volta che se ne ha bisogno. Il suo nome può essere costituito da lettere e numeri (ma questi non possono mai essere il primo carattere), e non può invece contenere simboli come `!`, `&` e spazi, che hanno un diverso significato specifico per la shell. L'operatore di assegnazione è il simbolo di uguale (`=`) non preceduto né seguito da spazi. Ad esempio, si può dare il comando: `poeta=Virgilio`. Qui «poeta» si dice *nome della variabile* e «Virgilio» *valore della variabile*.

Per indicare poi che un certo nome è una variabile e va interpretata, a esso si deve anteporre il simbolo del dollaro. In questo modo otterremo:

```
echo $poeta  
Virgilio
```

Le variabili possono essere usate per abbreviare dei comandi. Ad esempio, si può assegnare il nome `ldir` a un percorso assoluto completo come `/home/samiel/scritti/`. A questo punto, per copiarvi un file, basterà fare il comando: `cp testo2.txt $ldir`. Per verificare le variabili definite si usa il comando `set nome_variabile`, mentre per eliminarne una il comando `unset nome_variabile`.

Tali variabili sono però locali, la cui durata è cioè legata all'esecuzione della shell nella quale sono state definite. Inoltre, per valere anche per altri programmi, lanciati dalla shell stessa, devono essere esportate dopo essere state definite. Avremo perciò, ad esempio, il comando: `export poeta`. Verificarlo è molto semplice. Se definissimo all'interno di una shell per la variabile «poeta» il valore «Virgilio» e quindi, da questa shell, ne lanciassimo un'altra (una sottoshell), come `xterm`, il comando `echo $poeta` non produrrebbe alcun risultato. Se invece esportassimo quella variabile, allora la vedremmo funzionare anche in `xterm`.

18.9.3 Parametri posizionali

All'interno di uno script di shell, è possibile far assumere dei parametri dalla riga di comando o da un altro script facendo riferimento a delle variabili numeriche di chiamata. Poiché \$0 è il comando con cui è stato lanciato lo script, il primo parametro verrà memorizzato in una variabile chiamata 1 e invocata utilizzando \$1, e così via.

18.9.4 Variabili incorporate

Le variabili incorporate sono delle variabili speciali definite dal sistema, il cui valore non può essere modificato all'interno di un programma di shell. Esse servono per prendere delle decisioni all'interno di un programma.

18.9.5 Variabili d'ambiente

La bash ha delle variabili incorporate (*built-in*), dette variabili d'ambiente e definite nei file di configurazione di /etc (come /etc/profile), di /etc/skel (le quali risultano pertanto delle variabili globali) e nei file di configurazione presenti nella home dei vari utenti (le quali sono pertanto delle variabili locali, in un senso leggermente diverso a quello accennato sopra, perché relative non alla sessione di shell, ma per l'appunto al singolo utente), che vengono caricate ogni qualvolta l'utente si connette. Si utilizzano per ottenere informazioni come il tipo di sistema in esecuzione, la directory home e le shell utilizzate. Fra di esse troviamo:

HOSTNAME lo hostname della rete

HOME la collocazione della home dell'utente corrente

LANG la lingua predefinita del sistema

LD_LIBRARY_PATH il percorso in cui il linker dinamico cerca le librerie

PATH il percorso predefinito degli eseguibili

PWD la corrente directory di lavoro

SHELL nome e percorso della shell corrente

TERM il tipo di terminale utilizzato

USER l'utente connesso

Le variabili di ambiente si possono consultare coi comandi `env` o `printenv` `set`. Le variabili locali si possono consultare col comando `env`.

Fra l'altro, abbiamo visto come negli SlackBuild sia conveniente adottare delle variabili, definite all'inizio, per evitare di riscrivere varie volte gli stessi nomi. Se si deve ricompilare una versione aggiornata di un pacchetto, basta sostituire il numero di versione una sola volta, nella riga di assegnazione.

18.10 Modificare il prompt

Una delle variabili speciali della shell è il prompt, il cui nome è `PS1`. Secondo l'impostazione di /etc/profile, essa mostra `hostname` e `pwd`, cioè nome utente e directory di lavoro di default (la home dell'utente):

```
PS1=' \u@\h:\w\$ '
```

La `pwd` naturalmente cambia se l'utente si sposta in un'altra directory. Si veda infatti:


```
samiel@darkstar:~$
samiel@darkstar:~$ cd /etc
samiel@darkstar:/etc$
```

Il prompt di root è invece:

```
PS1='`hostname`:`pwd`# '
```

Il prompt può essere modificato da shell attribuendo un nuovo valore alla variabile PS1, col comando `PS1=nome_nuovo`. Ad esempio, digitando `PS1=eccomi` il prompt sarà, per l'appunto: `eccomi`. Se il nome contiene spazi, va racchiuso fra apici semplici, se contiene apici semplici (come un apostrofo), fra apici doppi. Per ottenere invece un testo speciale, nella variabile si inseriscono dei caratteri speciali, che richiedono la seguente sintassi:

```
PS1='\variabile simbolo_del_prompt' [più uno spazio]
```

Le variabili da passare sono le seguenti:

`\a` = fa suonare il campanello del sistema

`\d` = la data corrente

`\h` = lo hostname del sistema

`\n` = un carattere di riga nuova

`\t` = l'ora corrente nel formato 24 ore

`\@` = l'ora corrente nel formato 12 ore

`\u` = nome dell'utente

`\w` = la directory corrente

`\!` = il numero cronologico del comando all'interno della sessione corrente.

Naturalmente, queste modifiche valgono solo per la shell aperta, non si trasmettono alle altre e vanno perse quando si chiude la sessione. La variabile PS2 stabilisce il simbolo che mostra la shell quando il comando viene completato su più righe. Esso è, di default, `>`, come si evince anche da `/etc/profile`:

```
PS2='>'
```

18.11 Il path

La variabile `$PATH` è una delle più importanti, in quanto fissa il percorso (in genere, i percorsi) degli eseguibili. Se un utente cerca di lanciare un comando che corrisponde a un eseguibile non presente nel suo path, riceverà l'avviso:

```
samiel@darkstar:~$ ldconfig
-bash: ldconfig: command not found
```

Ci si può accertare dell'effettiva presenza dell'eseguibile col comando `whereis`, e quindi lanciarlo indicando il percorso completo. Normalmente, alcune directory come `/sbin` non si trovano nel path degli utenti perché il loro uso è riservato a root. Si può comunque rendere possibile anche agli utenti l'accesso a queste directory, aggiungendo il relativo path al loro `~/bash_profile`, nel modo seguente:

```
PATH=$PATH:/sbin
```

e quindi facendo leggere il `~/bash_profile` modificato col comando `source`.

18.12 Messaggi di errore

La shell tende a dare indicazioni precise al verificarsi di un errore. Ad esempio, il semplice comando: `ls /aaaaaa` (posto che tale directory non esista) produce il messaggio:

```
/usr/bin/ls: /aaaaaaaa: No such file or directory
```

Esso ci dice il programma richiamato col suo intero percorso, il nome del file, l'errore specifico. Di norma, se compaiono diversi errori, il più rilevante o quello di base è il primo. I messaggi di errore vanno distinti dai semplici avvisi («WARNING»). Questi ultimi indicano la presenza di qualcosa di errato, che però non impedisce il (quasi) regolare svolgimento del comando. Analizziamo alcuni errori fra i più comuni e le strategie di soluzione.

No such file or directory si è cercato di accedere a un file o a una directory che non esiste.

File exists si è cercato di creare un file o una directory che esiste già.

Not a directory, Is a directory si è cercato di usare un file come se fosse una directory o una directory come se fosse un file. Ad esempio, si crei il file *a* con `touch a`. Si cerchi quindi di creare il file *b* con `touch a/b`. Un messaggio ci avvertirà che *a/b* non è una directory. In questo caso è necessario interpretare il messaggio, perché in effetti è solo *a* a non essere una directory.

No space left on device si cerca di avviare un processo che occupa spazio (come quando si tenta di creare un file nuovo su di un dispositivo che non ha più spazio libero).

Permission denied si è cercato di effettuare un'operazione su un file o una directory su cui non si ha autorizzazione di accesso.

Operation not permitted si è cercato di terminare un processo di cui non si ha la proprietà.

Segmentation fault, Bus error il primo errore spesso deriva da un errore presente nel programma (dunque responsabilità del programmatore, non del sistema che sta eseguendo quel programma). Di norma l'errore di segmentazione deriva dal fatto che il programma ha cercato di accedere a una parte della memoria che non poteva utilizzare. L'errore di bus invece deriva dal fatto che si usa la memoria in modo non consentito.

18.13 Gli script di shell

Svariati comandi possono essere inseriti in sequenza in un file che ne determina l'esecuzione, detto script di shell (che qui comunque non analizzeremo in dettaglio, essendo un argomento complesso e non specifico di Slackware). I comandi di shell funzionano insomma come un programma, e a loro volta diverse centinaia dei comandi di GNU/Linux, come molti file di configurazione di Slackware, non sono che script di shell. Sebbene programmi complessi non possano essere redatti in questo modo, per i limiti oggettivi della shell e per la sua scarsa velocità, alcune istruzioni brevi vengono per l'appunto raccolte in file del genere, che presentano il vantaggio di non richiedere librerie aggiuntive. Essi sono file di puro testo, relativamente semplici da redigere, che, per essere eseguiti, vanno lanciati preceduti da `sh` e dunque col comando `sh nome_script` (talvolta specificando che lo si lancia dalla directory corrente, dunque `./`), oppure conferendo allo script il permesso di esecuzione (con `chmod a+x`). Le shell bash compatibili e le shell C compatibili obbediscono a sintassi diverse; per questo ogni script andrà aperto da una «istruzione bang» (`#!`) che controlla il tipo di shell per cui è stato redatto lo script. Ad esempio, tutti gli script di bash, per forzare la compatibilità con sh (la shell di Bourne), saranno aperti dalla riga:

```
#!/bin/sh
```

mentre tutti gli script di shell C dalla riga:

```
#!/bin/csh
```

Un esempio molto elementare di script per bash che chiamiamo qui `myalias` è il seguente:

```
#!/bin/sh
alias ll='ls -l'
alias ldir='ls -aF'
alias copy='cp'
```

che altro non è, se non l'assegnazione di alcuni alias. Tale script può essere eseguito volta per volta (mediante i comandi `sh myalias` se non è eseguibile, oppure `./myalias` se è eseguibile). In alternativa, opportunamente collocato, esso attiverà gli alias a ogni avvio del sistema.

18.14 Utenti e processi

18.14.1 Visualizzare gli utenti e il loro accesso al sistema

Il comando `whoami` visualizza il nome dell'utente con cui al momento si è effettuato il login (che può essere diverso da quello che ha effettuato l'accesso se si usa `su`).

Il comando `id` non si limita a visualizzare il nome dell'utente con cui al momento si è effettuato il login, ma aggiunge tutti i gruppi (per numero e per nome) di cui l'utente è membro.

Il comando `who` visualizza il nome di tutti gli utenti che sono connessi al sistema. Indica inoltre il terminale a cui sono connessi e l'ora di accesso:

```
root@darkstar:~# who
samie1  tty1    Apr 4 10:29
sabina  tty3    Apr 4 12:40
root    tty2    Apr 4 12:38
```

Questo comando è limitato agli utenti presenti sul sistema locale, e non include quelli che hanno effettuato l'ingresso dalla rete, individuabili col comando `finger`.

Il comando `w` è analogo a `who`, ma fornisce una maggiore quantità di dettagli:

```
root@darkstar:~# w
12:41:39 up 2:21, 3 users, load average: 1,06, 0,53, 0,20

USER      TTY      FROM      LOGIN@    IDLE      JCPU      PCPU      WHAT
samie1    tty1     -         10:29     2:11m    0.04s    0.00s    /bin/sh  /bin/sh
sabi      tty3     -         12:4     1:27     0.02s    0.00s    /bin/sh  /usr/X1
root      tty2     -         12:8     2:34     0.03s    0.00s    emacs
```

Nella prima riga si leggono l'ora corrente del sistema, da quanto esso è attivo (*up*), il numero degli utenti. Nelle righe successive sono elencati gli utenti connessi, il terminale che stanno adoperando, la sessione X se presente, l'ora della connessione o login, il tempo per cui sono rimasti inattivi, i programmi che stanno utilizzando.

Il comando `last` elenca gli utenti che si sono connessi di recente al sistema, mentre seguito dal nome di un utente visualizza data e ora del suo ultimo accesso. Esso trae i dati dal file `/var/log/wtmp`, che indica nell'ultima riga, assieme all'ora e alla data da cui esso parte.

18.14.2 Visualizzare i processi del sistema

Il comando `ps` indica i processi attivi nel sistema; se invocato da un utente, mostra soltanto i processi attivati da questo. Senza opzioni, fornisce solo delle informazioni di base.

```
samiel@darkstar:~$ ps
```

PID	TTY	TIME	CMD
7305	pts/2	00:00:00	bash
7316	pts/2	00:00:00	ps

Più in dettaglio, la voce PID (acronimo di «Process IDentifier) precisa il numero unico identificativo del processo stesso. La voce TTY mostra il terminale su cui stanno girando i processi; se sono attivati tutti sullo stesso terminale, ogni voce darà lo stesso tty. La voce TIME indica il tempo di CPU impiegato dal processo per partire; poiché in genere sono presenti nel sistema molti processi che lavorano tutti allo stesso tempo, ciascuno di essi impegna solo una piccola porzione del tempo della CPU. Un tempo particolarmente lungo di avvio, nell'ordine dei minuti, è un dato sospetto e in genere sintomo di qualche problema del sistema. Infine, la voce CMD mostra l'eseguibile da cui dipende il processo.

Per ottenere una completa lista dei processi attivi nel sistema, si aggiungono al comando `ps` ulteriori opzioni: l'opzione `[-]a` mostra tutti i processi di tutti gli utenti del sistema, tranne quelli che non sono associati a nessun terminale e i cosiddetti «group leaders» (un *group leader* è il primo membro di una serie di processi correlati). L'opzione `[-]u` mostra tutti i processi associandoli ai vari utenti, indicati con l'ID o col nome (si tratta di un output «user-oriented»). L'opzione `[-]x` mostra tutti i processi che non sono associati a nessun terminale, come i demoni (e infatti nella colonna TTY invece del terminale apparirà un punto di domanda). Vediamo alcuni esempi.

```
samiel@darkstar:~$ ps -ax
```

PID	TTY	STAT	TIME	COMMAND
1	?	S	0:00	init [3]
2	?	SN	0:00	[ksoftirqd/0]
3	?	S<	0:00	[events/0]
4	?	S<	0:00	[khelper]
9	?	S<	0:00	[kthread]
.				
.				
.				
4919	tty1	Ss	0:00	-bash
4920	tty2	Ss+	0:00	/sbin/agetty 38400 tty2 linux
4921	tty3	Ss+	0:00	/sbin/agetty 38400 tty3 linux
4922	tty4	Ss+	0:00	/sbin/agetty 38400 tty4 linux
4923	tty5	Ss+	0:00	/sbin/agetty 38400 tty5 linux
4924	tty6	Ss+	0:00	/sbin/agetty 38400 tty6 linux
5241	tty1	S+	0:00	/bin/sh /usr/X11R6/bin/startx
.				
.				
.				
6091	pts/1	Ss+	0:00	-bash
6105	?	R	0:00	konsole [kdeinit] konsole -ls
6107	pts/2	Ss	0:00	-bash

```
6119 pts/2 R+    0:00 ps ax
```

Qui notiamo in più la colonna *STAT*, che indica lo stato di un processo. Le possibilità sono:

D: indica i processi che sono «dormienti» e che non possono essere attivati con nuovi input

I: indica i processi inattivi («idle»)

N: indica i processi a bassa priorità

R: indica i processi attualmente in atto («running»)

S: indica i processi «dormienti» in attesa di nuovi input («sleeping»)

T: indica i processi interrotti («traced»)

Z: indica i processi defunti, cioè quelli di cui è morto il processo padre, che però non è riuscito a distruggerli («killarli») in modo appropriato; essi sono detti «zombie».

A quest'ultimo proposito, ricordiamo che l'opzione `-l` aggiunge informazioni su quali processi padre hanno avviato ciascun processo figlio.

Per visualizzare graficamente in modo più chiaro la dipendenza dei processi, è utile il comando `pstree`.

La colonna *COMMAND* indica il nome del comando che ha avviato il processo.

Informazioni ancora più dettagliate si ottengono col comando `ps [-]aux`:

```
samiel@darkstar:~$ ps aux
USER      PID  %CPU  %MEM  VSZ   RSS  TTY   STAT  START  TIME  COMMAND
root       1    0.0   0.0   668   228  ?     S      22:11  0:00  init [3]
root       2    0.0   0.0    0     0  ?     SN     22:11  0:00  [ksoftirqd/0]
root       3    0.0   0.0    0     0  ?     S<     22:11  0:00  [events/0]
root       4    0.0   0.0    0     0  ?     S<     22:11  0:00  [khelper]
root       9    0.0   0.0    0     0  ?     S<     22:11  0:00  [kthread]
.
.
.
root      3351  0.0   0.0   1528   468  ?     Ss     22:12  0:00  /usr/sbin/gpm -m
mysql     4848  0.0   1.4  317756 14808  ?     S      22:12  0:00  /usr/libexec/mysq
mysql     4849  0.0   1.4  317756 14808  ?     S      22:12  0:00  /usr/libexec/mysq
mysql     4850  0.0   1.4  317756 14808  ?     S      22:12  0:00  /usr/libexec/mysq
mysql     4861  0.0   1.4  317756 14808  ?     S      22:12  0:00  /usr/libexec/mysq
samiel    4919  0.0   0.1   3344   1892  tty1  Ss     22:12  0:00  -bash
.
.
.
samiel    6089  0.3   1.6   29148 16780  ?     S      23:18  0:00  konsole [kdeinit]
samiel    6091  0.0   0.1   3344   1888  pts/1  Ss+    23:18  0:00  -bash
samiel    6105  0.4   1.6   29148 16792  ?     R      23:19  0:00  konsole [kdeinit]
samiel    6107  0.0   0.1   3344   1892  pts/2  Ss     23:19  0:00  -bash
samiel    6122  0.0   0.0   2476    844  pts/2  R+     23:21  0:00  ps aux
```

Si leggono qui ulteriori notizie come sull'utente che ha avviato il processo, sulle risorse di sistema che esso sta occupando: le colonne *%CPU*, *%MEM*, *VSZ*, *RSS* indicano rispettivamente la percentuale di potenza del processore utilizzata nell'ultimo minuto, la percentuale di memoria usata nell'ultimo minuto, la

dimensione della memoria virtuale totale in KB (Virtual SiZe), i KB di memoria reale occupata (Resident Set Size); la voce *START* indica il momento in cui il processo è stato avviato.

Quando le righe sono così lunghe che eccedono lo spazio orizzontale dello schermo, l'opzione *-w* consente di farle andare a capo.

Per visualizzare particolari processi il cui nome contiene specifiche stringhe, si convoglia l'output di *ps* a *grep*. Ad esempio, per visualizzare i processi i cui comandi contengono un riferimento alla directory */sbin*, il comando sarà: `ps aux | grep sbin`, mentre per vedere quale processo corrisponde a un determinato PID il comando sarà: `ps -p nn`, dove *nn* è il PID del processo in questione.

18.14.3 Il comando top

Il comando *top* visualizza i processi del sistema ordinati secondo la richiesta di risorse del sistema. I dati vengono aggiornati di continuo. Per uscire dal programma si preme *q*. Tale comando visualizza inoltre alcune informazioni sul tempo di funzionamento del sistema e sull'uso della memoria che si possono ottenere anche separatamente con i comandi *uptime* e *free*.

18.14.4 Gestione dei processi

Visualizzare i jobs

Per ottenere l'elenco dei processi si usa il comando *jobs*, che ne indica il numero (fra quadre) e lo stato, cioè se sono in esecuzione («running») o fermi («stopped»). Il segno *+* indica che il job è in fase di attuazione, mentre il segno *-* indica il prossimo job che verrà eseguito.

Mettere un job in background

Ogni processo attivo può essere messo in background per liberare la shell ed eseguire altri programmi; ciò è utile per liberare la shell quando il processo sia piuttosto lungo. È possibile inoltre interrompere l'esecuzione di un comando prima della sua conclusione o sospenderla per riavviarla in un secondo momento dal punto in cui era stato interrotta. In ogni momento sono attivi i processi di sistema, mentre ogni comando impartito dagli utenti viene chiamato job, anche se è a tutti gli effetti un ulteriore processo.

Per mettere in background un job si inserisce il carattere *&* dopo il nome del comando. Ad esempio, per cercare un file mettendo la ricerca in background, avremo un comando che rilascerà un output, come segue:

```
bash-2.00# find / -name prova.txt &
[2] 2035
```

Il primo numero, fra parentesi quadre, è il numero di job dell'utente; il secondo numero è il numero che assume il processo nel sistema (il suo PID).

Si possono mettere più lavori in background, con la sintassi: `comando1 [opzioni] [argomento] &; comando2 [opzioni] [argomento] &`.

Se il processo è già attivo, può essere messo in background col comando *bg*.

La shell comunica con il messaggio «Done» la conclusione del job in background solo quando il job attualmente attivo è terminato, per non interromperlo; se invece si vuole essere avvisati subito e comunque della sua conclusione, indipendentemente dal fatto di star effettuando altre operazioni, si utilizza il comando *notify*, passandogli come argomento il numero del job preceduto dal simbolo di percentuale, ad esempio `notify %2`.

È inoltre possibile far passare un comando in via di attuazione, e cioè in foreground, in background, o al contrario collocare in background un job attualmente in foreground. Nel primo caso si usa il comando

`fg`: se c'è un solo job in background, esso viene portato automaticamente in foreground, mentre se ve ne sono più di uno, è necessario specificare il numero del job preceduto dal simbolo %, ad esempio `fg %4`. Il comando opposto è `bg`, usato di solito per riprendere i job sospesi. Questo comando accetta come argomento sia il numero del job, sia quello del processo.

Sospendere un job

La sospensione di un job si ottiene col comando **Ctrl+z**, che lascia aperta la possibilità di riprenderlo in seguito col comando `fg`, riportandolo cioè in foreground. La shell visualizza una riga che indica il numero del job sospeso (fra parentesi quadre), la voce STOPPED e il nome del comando relativo a quel job.

Terminare un job

Per arrestare un job in foreground prima del suo completamento, si usa il comando **Ctrl+c**, che fa uscire dal programma. Si può anche adottare il comando `kill`, seguito dal numero del job sempre preceduto da %, oppure dal numero del processo.

Capitolo 19

Tips & tricks

Sono qui raccolte alcune delle tante procedure per settaggi particolari, alcune tipiche di Slackware, altre dotate di più generale validità.

19.1 Installare il kernel 2.6

In attesa che Slackware adotti di default il kernel 2.6, è comunque possibile installarne una versione precompilata, per chi non volesse affrontare manualmente la sua compilazione. Slackware 11.0 presenta addirittura due kernel di questa serie: il 2.6.17.13 e il 2.6.18.

19.1.1 Installare il kernel 2.6 dopo l'installazione

Slackware 11.0 adotta di default il kernel 2.4.33.3. Tuttavia nella directory `/extra/linux-2.6.17.13` è presente, come abbiamo anticipato, un kernel precompilato della serie 2.6.17.13, mentre nella directory `/testing/packages/linux-2.6.18` è presente anche un kernel precompilato della serie 2.6.18, detti entrambi *kernel-generic*. La loro installazione è estremamente semplice; basta infatti effettuarla con un comando come: `installpkg kernel-generic-2.6.17.13-i486-1.tgz`.

È opzionale installare, sempre con `installpkg`, anche i moduli (`kernel-modules-2.6.17.13-i486-1.tgz`) e gli header (`kernel-headers-2.6.17.13-i486-1.tgz`). Non si deve dimenticare che l'installazione dei moduli è in genere un'operazione da compiere senz'altro, perché in caso contrario potrebbero verificarsi dei malfunzionamenti del sistema (non funzionerebbero tutti i dispositivi governati da moduli), mentre quella degli header è più rischiosa, perché potrebbe causare dei problemi nella eventuale compilazione di pacchetti. Infatti è buona norma non installare degli header più recenti della versione con cui sono compilate le `glibc`, nel caso di Slackware 11.0 la 2.3.6.

Una volta effettuata l'installazione e prima di riavviare il sistema, ci sono due operazioni da effettuare. Innanzitutto creare un `initrd`, in modo da caricare i moduli opportuni per il file system prima di avviare il nuovo kernel. Infatti i vari file system sono impostati nel kernel precompilato in forma modulare, per consentire qualsiasi scelta evitando allo stesso tempo un incremento delle dimensioni del kernel, mentre quello della partizione di root dev'essere impostato staticamente per evitare un kernel panic che blocca il caricamento del sistema. Per ovviare a tale conseguenza si ricorre per l'appunto a `initrd`. L'unico caso in cui non serve un `initrd` è se il file system della partizione di root è EXT2 e il disco fisso è IDE: infatti il kernel precompilato li supporta entrambi nativamente.

La procedura è leggermente diversa a seconda del file system adottato nella partizione di root. Se si tratta di EXT3, si lancerà il comando: `mkinitrd -c -k 2.6.17.13 -m jbd:ext3 -f EXT3 -r /dev/hd xn` ,

dove x è il disco fisso e n è il numero della partizione di root. Se invece il file system è Reiserfs, allora il comando risulta essere: `mkinitrd -c -k 2.6.10 -m Reiserfs /dev/hd a n` .

Serve un po' di lavoro in più se il disco fisso dove risiede la partizione di root è SATA, perché bisogna far sì che `initrd` carichi anche il relativo supporto. In questo caso, poiché i driver sono tutti presenti nel pacchetto dei moduli, bisogna lanciare un comando del tipo: `mkinitrd -c -k 2.6.17.13 -m libata:ata_piix:jbd:ext3 -f ext3 -r /dev/sda2`, posto qui che il file system sia EXT3, la partizione di root sia `/dev/sda2` e che il modulo che supporta il controller del disco fisso sia `piix:jbd`. I moduli per i dischi fissi SATA sono tutti presenti in `/lib/modules/2.6.17.13/kernel/drivers/scsi`.

Dopo aver costruito `initrd`, si deve aggiornare il file `/etc/lilo.conf` aggiungendo nella sezione «# Linux bootable partition config begins» le seguenti righe:

```
image = /boot/vmlinuz-generic-2.6.17.13
initrd = /boot/initrd.gz
root = /dev/hd $a$  $n$ 
label = Linux-2.6
read-only
```

Al solito, si deve lanciare `lilo [-v]` per aggiornare l'MBR. Al riavvio, sarà possibile scegliere uno dei due kernel. Questo procedimento installa nella directory `/boot` i nuovi `vmlinuz`, `config` e `System.map`, ma non anche i sorgenti (che si collocano in `/usr/src`): di conseguenza, non sarà possibile ricompilare questo kernel senza installare separatamente, in un secondo momento, i sorgenti stessi.

19.1.2 Installare il kernel 2.6 al momento dell'installazione

Slackware 11.0 ci dà la possibilità di installare direttamente il kernel 2.6.17.13 al momento dell'installazione originaria (non così invece per il kernel 2.6.18). A tal fine, alla prima richiesta del sistema, ovvero quando si deve scegliere il kernel con cui avviare, si dovrà scrivere al prompt: `huge26.s`. Questo kernel supporta nativamente anche i dischi SATA e SCSI. Tuttavia richiede un intervento successivo, perché seguendo tale procedura non installa in aggiunta né i moduli né, tanto meno, gli header. Terminate tutte le operazioni di installazione e di prima configurazione, si dovrà pertanto montare il CD e installare con `installpkg` i pacchetti dei moduli ed eventualmente gli header. Non è invece necessario installare i relativi driver di `Alsa`, perché il supporto audio è già incluso nel kernel; tuttavia, l'installazione degli header fornirà la directory `/usr/include/sound` usata per compilare le applicazioni di `Alsa`.

19.1.3 Installare un doppio kernel

È possibile (e, quando si procede con la compilazione di un nuovo kernel, consigliabile) mantenere il vecchio. In questo caso, in `/etc/lilo.conf` dovranno essere presenti entrambi, come dal seguente esempio:

```
image = /boot/vmlinuz-2.6.17.13
  root = /dev/sda2
  label = Slackware-2.6
  read-only
image = /boot/vmlinuz-2.4.33.3
  root = /dev/sda2
  label = Slackware-2.4
  read-only
append="hda=ide-scsi"
```

Se si installano i driver per l'accelerazione grafica, si dovrà seguire una procedura specifica perché essi siano operanti con entrambi i kernel.

Nel caso di una scheda grafica ATI si effettuerà l'installazione tradizionale ripetendola per entrambi i kernel.

Nel caso di una scheda grafica Nvidia si effettuerà l'installazione tradizionale con uno dei due kernel e si ripeterà col secondo aggiungendo l'opzione `-K`, di modo che il comando per lanciare l'installer risulterà: `sh NVIDIA-Linux-x86-1.0-8756-pkg1.run -K`. Con tale procedura, infatti, l'installer compila solo il modulo del kernel e non, in aggiunta, quello per X.

19.2 Velocizzare l'avvio di Slackware

All'avvio il programma carica numerose componenti e compie delle verifiche che, in genere, sono molto veloci. Tuttavia si rallenta per alcuni secondi nel aggiornare con `ldconfig` eventuali nuove librerie, per creare con `fc-cache` il database dei caratteri e per settare con `hotplug` le periferiche hardware. Normalmente i primi due controlli sono inutili, se non si sono fatte recenti nuove installazioni nel sistema. Essi si possono pertanto eliminare commentando le seguenti righe di `/etc/rc.d/rc.M`:

```
# Update all the shared library links:
# if [ -x /sbin/ldconfig ]; then
#   echo "Updating shared library links: /sbin/ldconfig"
#   /sbin/ldconfig
# fi
```

e

```
# Update the X font indexes:
# if [ -x /usr/X11R6/bin/fc-cache ]; then
#   echo "Updating X font indexes: /usr/X11R6/bin/fc-cache"
#   /usr/X11R6/bin/fc-cache
# fi
```

(dove si vedono già le righe commentate).

Sarà però necessario lanciare sempre `ldconfig [-v]` come root dopo aver installato nuove librerie o un nuovo programma e `fc-cache` dopo aver installato nuovi caratteri.

È invece assai comodo poter contare su `hotplug`. In questo caso, ma anche nei due precedenti, una soluzione alternativa consiste nel far eseguire i programmi in background. Per questo, invece di commentare tutta la sezione, le righe dove sono indicati gli eseguibili andranno messi per l'appunto in background, ricorrendo a `&`.

Nel caso di `hotplug`, le righe pertinenti di `/etc/rc.d/rc.M` sono:

```
if cat /proc/mounts | grep -wq sysfs ; then
  if ! grep -wq nohotplug /proc/cmdline ; then
    if [ -x /etc/rc.d/rc.udev ]; then
      if [ -d /dev/.udev/failed ]; then
        for i in /dev/.udev/failed/*; do
```

```

        echo "add" > "${i}/uevent"
    done
fi
elif [ -x /etc/rc.d/rc.hotplug ]; then
    . /etc/rc.d/rc.hotplug start &
fi
fi
elif [ -x /etc/rc.d/rc.hotplug -a -w /proc/sys/kernel/hotplug ]; then
    if ! grep -wq nohotplug /proc/cmdline ; then
        . /etc/rc.d/rc.hotplug start &
    fi
fi
fi

```

dove si vede già la riga col comando messo in background. Allo stesso modo la riga:

```
/sbin/ldconfig
```

andrà cambiata in:

```
/sbin/ldconfig &
```

e infine la riga:

```
/usr/X11R6/bin/fc-cache
```

diventerà:

```
/usr/X11R6/bin/fc-cache &
```

19.3 Lanciare programmi all'avvio

Per far sì che alcuni programmi siano lanciati automaticamente all'avvio, è necessario editare il file `/etc/rc.d/rc.local` aggiungendo il comando completo, e cioè comprensivo del percorso assoluto del file.

19.4 Accendere il tastierino numerico all'avvio

Perché il tastierino numerico sia acceso quando ci logghiamo nel sistema dobbiamo editare il file `/etc/rc.d/rc.local` (che contiene i comandi definiti dall'utente che il computer deve compiere all'avvio). Aggiungiamo dunque in questo file con un editor qualsiasi le seguenti righe:

```

#accensione del tastierino numerico all'avvio
for tty in /dev/tty[1-9]*; do
    setleds -D +num < $tty
done

```

È necessario fare attenzione alla codifica da adottare, qualora si usi un editor che supporta varie codifiche (come Kwrite o Kate), nella fattispecie Unicode (UTF-8), o il file non riuscirà ad essere letto, generando all'avvio un messaggio di errore («command not found»). La procedura appena descritta vale per la shell di login e per i terminali non grafici. Per avere il tastierino acceso all'avvio nei vari Window Manager, è necessario impostare la relativa preferenza (in KDE da **Centro di controllo – Periferiche – Tastiera**, alla voce **BlocNum all'avvio di KDE** bisogna selezionare **Acceso**).

19.5 Rimuovere fortune

Se i messaggi che ricevete all'avvio del programma o della shell vi divertono, continuate a leggerli. Se ritenete che vi facciano perdere tempo o attenzione nei confronti di ciò che intendete fare, potete rimuoverli anche senza eliminare i pacchetti modificando i permessi di `/etc/profile.d/bsd-games-login-fortune.sh` e di `/etc/profile.d/bsd-games-login-fortune.csh` col comando: `chmod a-x` (in una sola mossa: `chmod a-x /etc/profile.d/bsd*`). In questo modo si impedisce loro di essere eseguiti.

19.6 Modificare i caratteri della console

Per modificare i caratteri della console, oltre a operare sul menu della console stessa, è possibile lanciare il comando: `setconsolefont nome_font`, oppure: `setfont -v nome_font`, oppure ancora editare manualmente il file `/etc/rc.d/rc.font` inserendo dopo la stringa `setfont -v` il nome di uno dei caratteri presenti in `/usr/share/kbd/consolefonts`. Il solo comando `setconsolefont` avvia lo script di configurazione raggiungibile anche da `pkgtool`.

A volte alcune shell (come Konsole di KDE) possono dare l'errore:

```
«Carattere "-misc-console-medium-r-normal--16-160-72-72-c-80-iso10646 -1"
non trovato».
```

In questo caso la soluzione consiste nella seguente procedura:

- a) dal **Centro di controllo** di KDE entrare nell'installatore di caratteri
- b) cliccare sull'icona **Aggiungi caratteri**
- c) andare nella directory `/opt/kde/share/fonts` e aggiungere i file di estensione `.pcf.gz`
- d) lanciare da console il comando: `fc-cache` per aggiornare il database dei font
- e) avviare Konsole e scegliere dal menu **Impostazioni – Caratteri – Linux** il carattere, ora reperibile.

19.7 Eliminare il messaggio «mySQL ended»

Per eliminare il messaggio «mySQL ended» all'avvio del programma bisogna abilitarlo: si tratta della prima operazione da fare se si vuole adoperare questo database. È necessario in prima istanza scegliere uno dei quattro file `/etc/my-huge.cnf`, `/etc/my-large.cnf`, `/etc/my-medium.cnf`, `/etc/my-small.cnf` e copiarlo sempre in `/etc` rinominandolo `/etc/my.cnf`. La scelta dipende dalle dimensioni che intendiamo attribuire al nostro database. Quindi bisogna lanciare in sequenza i seguenti comandi: 1) su `mysql -c mysql_install_db`; 2) `cd /var/lib`; 3) `chown -R mysql mysql \;`; 4) `chgrp -R mysql mysql \`. In tal modo mySQL viene attivato. Eccezione dalla presente trattazione l'uso di questo database.

19.8 C'è post@ per te

Al primo avvio da root, si nota che, al prompt, compare l'avviso: «You have new mail». Si tratta di due scritti indirizzati da P. J. Volkerding all'utente root.

Il primo ci invita a registrarci come utenti GNU/Linux presso il sito <http://counter.li.org>, una sorta di anagrafe per quantificare il numero degli utenti. L'iscrizione, che non comporta alcun obbligo, consente di registrare sia l'utente sia il suo computer a fini puramente statistici.

Il secondo scritto affronta diversi problemi di configurazione dell'hardware: porte parallele e seriali, masterizzatori, stampanti, mouse e modem.

Come leggere queste due e-mail? Ci sono due possibilità. Dal momento che esse sono state redatte con pine (un client e-mail da riga di comando), possiamo richiamare il programma stesso col comando: `pine`. Se è la prima volta che lo avviamo, riceveremo un messaggio di benvenuto da cui usciremo col tasto **e** e quindi col cursore (oppure premendo il tasto **i**) andremo alla voce `MESSAGE INDEX`, dove troveremo i due messaggi in questione. Si esce infine da pine premendo il tasto **q** e confermando. In alternativa, possiamo richiamare col comando: `mail` l'omonimo programma testuale di posta elettronica. Con i tasti **1** e **2** si richiameranno le due e-mail e col tasto **Invio** se ne scorrerà il testo.

19.9 Regolare la data e l'ora

Il fuso orario si può impostare con l'utilità pseudo-grafica `timeconfig`, che consente di scegliere fra UTC e ora locale. La data e l'ora locale si impostano in console con `date`, e più precisamente col comando: `date --set=MMGGhhmm`. L'opzione `-u` dà l'ora UTC.

Si può avere sempre l'ora esatta sincronizzandosi mediante `ntpdate` con un server SNTP. Per evitare di effettuare a mano l'operazione, basta aggiungere la stringa `/usr/sbin/ntpdate nome_server` al file `/etc/rc.d/rc.local`, eventualmente con l'ulteriore stringa `>/dev/null` per non visualizzare a schermo tutto lo svolgersi della peraltro rapida procedura: a ogni avvio del sistema, verrà effettuata la sincronizzazione.

Gli elenchi dei server a cui connettersi si individuano facilmente grazie ai motori di ricerca. Un elenco di server europei con cui sincronizzarsi si legge all'indirizzo: <http://www.pool.ntp.org/zone/europe>. Qui ricordiamo, fra i tanti: `time.ien.it`, `time.nist.gov`, `ntp2.ien.it`, `ntp.univ-lyon1.fr`.

È anche possibile effettuare la sincronizzazione non *una tantum*, ma costantemente, mediante `ntpd`, che si vale del file di configurazione `/etc/ntp.conf`. Ciò tuttavia è utile in particolar modo per i server, e pone alcuni problemi di impostazione e di sicurezza. Questo argomento eccede dalla presente trattazione.

L'orologio di sistema si imposta con il comando: `hwclock --set --date=gg/mm/aa hh:mm:ss`. L'opzione `-u` dà l'ora UTC.

19.10 Tradurre le man pages in italiano

Le man pages sono la prima, fondamentale risorsa per avere informazioni sui comandi di GNU/Linux. Una comunità italiana, il Progetto Pluto, ha realizzato la traduzione italiana della più gran parte delle man pages, per aiutare i non anglofoni, reperibile all'indirizzo: <http://ftp.pluto.linux.it/pub/pluto/ldp/man>; si scarica così il file denominato `man-pages-it-2.22.tar.gz` o superiore. Lo si può quindi installare con la tradizionale procedura di compilazione, o creando un `.tgz`. Le pagine tradotte saranno collocate nella directory `/usr/man/it`. Il loro utilizzo presuppone l'italianizzazione di Slackware che abbiamo descritto sopra. Unico limite: la traduzione non è completa né sempre aggiornata rispetto alle pagine man originali.

19.11 Installare caratteri

Per installare in Slackware nuovi caratteri TTF o Type1, ci sono svariate procedure. Il vecchio sottosistema **font/x** è ormai deprecato (abbandonato nelle ultime versioni di Slackware) e sostituito dal nuovo sottosistema **Fontconfig**, che consente prestazioni di visualizzazione decisamente migliori (come un avanzato anti-aliasing). Per migliorare l'anti-aliasing, serve una versione di Freetype in cui esso stesso sia abilitato. Tuttavia il sottosistema Fontconfig non funziona ancora per alcune applicazioni come OpenOffice e Abiword, che usano una propria tecnologia di font rendering. Qui consideriamo in prevalenza la procedura di installazione globale, cioè un'installazione compiuta da root che installerà caratteri che saranno disponibili per tutti gli utenti. È comunque possibile effettuare anche un'installazione locale, da parte di un singolo utente, col risultato che i caratteri aggiunti saranno disponibili solo per lui.

19.11.1 Il sottosistema font/x

Col sottosistema font/x la procedura di aggiunta di caratteri è la seguente:

- 1) copiare i caratteri in una directory creata appositamente col comando: `mkdir /usr/share/fonts/locale` e quindi collocarsi in questa directory.
- 2) Aggiungere il percorso la directory dei caratteri al server di caratteri col comando: `chkfontpath --add /usr/share/fonts/local`.
- 3) Generare la lista dei caratteri col comando:
`ttmkfdir -d /usr/share/fonts/local -o /usr/share/fonts/local/fonts.scale`.
- 4) Generare il file `font.dir` col comando: `mkfontdir`. A questo punto dovremmo trovare nella directory, oltre ai caratteri copiati, due nuovi file: `fonts.scale` e `font.dir`.
- 5) Riavviare il server dei caratteri col comando: `service xfs restart` o riavviare il server grafico.

19.11.2 Il sottosistema fontconfig

Col recente sottosistema fontconfig la procedura di aggiunta globale di caratteri è molto più semplice: è sufficiente copiare i caratteri in una delle directory definite nelle opzioni di fontconfig, che è possibile trovare elencate nel file `/etc/fonts/fonts.conf`, in apertura, alla voce «Font directory list». Va infine lanciato il comando: `fc-cache /dir/dir/dir_dei_font` per aggiornare la cache delle informazioni dei caratteri.

19.11.3 L'installazione individuale

Per aggiungere i caratteri a un utente individuale, basta copiare i nuovi caratteri nella directory `.fonts/` nella home directory dell'utente e quindi lanciare il comando: `fc-cache -fv /dir_dei_caratteri`. Il file di configurazione dei caratteri è `/etc/fonts/fonts.conf`, che però non può essere modificato manualmente.

19.11.4 Aggiungere caratteri con KDE

Poiché fontconfig è usato automaticamente per le applicazioni programmate usando il toolkit grafico Qt 3 o GTK+ 2, con KDE l'aggiunta di nuovi caratteri può essere effettuata in modo molto semplice dal browser Konqueror trascinando nella directory `fonts/Personale` e relative subdirectory i file dei caratteri da copiare. Il metodo più semplice resta comunque valersi dell'utilità **Installatore dei tipi di carattere** dal **Centro di controllo** di KDE.

19.12 Stampare con OpenOffice o con StarOffice

Vi sono applicazioni che, una volta configurata la stampante, la trovano subito. In qualche altro caso è necessario lavorare un po' di più, come con OpenOffice (o con StarOffice). Poniamo di aver già configurato la nostra stampante con CUPS, per cui il suo driver è già stato decompresso e si trova in `/etc/cups/ppd`. Seguiamo allora i seguenti passaggi:

- a) apriamo il menu **Gestione stampanti** di OpenOffice
- b) premiamo il tasto **Nuova stampante**
- c) nella nuova finestra mettiamo la spunta ad **Aggiungi una nuova stampante**
- d) premiamo il tasto **Avanti**
- e) se nella finestra **Selezionate un driver** esso è presente, basta selezionarlo
- f) se nella finestra **Selezionate un driver** esso non è presente, premiamo il tasto **Importa**
- g) nella finestra **Cartella driver** selezioniamo **Sfoglia** e andiamo in `/etc/cups/ppd`, quindi premiamo il tasto **Seleziona**
- h) nella finestra successiva, nell'elenco **Selezione driver** troviamo il nome del nuovo driver; selezioniamolo e premiamo il tasto **Ok**
- i) Nella finestra **Selezionate una riga di comando** inseriamo la stringa `lpr-cups` e premiamo **Avanti**
- l) A questo punto premiamo il pulsante **Crea**, eventualmente dopo aver spuntato la casella **Imposta come stampante predefinita**.

Nel caso non avessimo installato CUPS, sarà necessario decomprimere il driver situato nella directory `/usr/share/ppd/marca_della_stampante` e quindi installarlo seguendo la procedura di installazione appena indicata.

19.13 Aprire link in Mozilla

Se dobbiamo aprire più finestre di Mozilla da alcuni links presenti in un documento di OpenOffice (o di StarOffice), ci troveremo di fronte a una situazione molto scomoda. Mozilla infatti (ma da questo problema non è affetto Firefox) aprirà una sessione diversa per ogni link, richiedendo ogni volta un profilo utente diverso. Ma è affatto inusuale che un utente si crei una molteplicità di profili diversi. Per ovviare all'inconveniente dobbiamo redigere un script del tipo:

```
#!/bin/bash
if ! mozilla -remote "openurl($1,new-window)"
then
  exec mozilla $1
fi
```

salvarlo in una directory qualsiasi, chiamarlo a piacere e renderlo eseguibile con il comando `chmod a+x`. Infine dobbiamo settarlo come default browser in OpenOffice o in StarOffice al menu **Strumenti – Opzioni – Programmi ausiliari – Attiva i collegamenti con HTTP**, indicando ovviamente l'intero percorso dello script.

19.14 Vedere GNU/Linux da Windows

Per rendere accessibili le partizioni GNU/Linux del disco fisso da Windows, e cioè per poterne leggere il contenuto ed effettuare lo spostamento o la copia dei file, esistono alcune comode utilità.

19.14.1 Vedere le partizioni EXT2 ed EXT3

Esistono svariate utilità che consentono di vedere le partizioni EXT2 ed EXT3. Ricordiamo qui innanzitutto **explore2fs**, scaricabile all'indirizzo <http://uranus.it.swin.edu.au/~jn/linux/explore2fs.htm>. Esso si presenta come una sorta di comoda «Gestione risorse» di Windows con cui è possibile non solo vedere le partizioni GNU/Linux dalla partizione di Windows, ma anche copiarvi dei file da Linux.

Altre utilità analoghe, ma con caratteristiche lievemente diverse, sono le seguenti:

ext2ifs, scaricabile all'indirizzo: <http://uranus.it.swin.edu.au/jn/linux/ext2ifs.htm>

ext2fsd, scaricabile all'indirizzo: <http://ext2fsd.sourceforge.net/index.htm>

gli **fs-driver**, scaricabili all'indirizzo: <http://www.fs-driver.org>.

Si tratta di driver che consentono di montare le partizioni EXT2 ed EXT3 su Windows assegnando loro una lettera di unità, per cui risultano visibili direttamente dalla Gestione risorse; la seconda e la terza utilità consentono anche la scrittura.

19.14.2 Vedere le partizioni Reiserfs

Per vedere le partizioni formattate in Reiserfs si può fare ricorso a **rfstool**, un'utilità di sola lettura a riga di comando scaricabile all'indirizzo: <http://p-nand-q.com/download/rfstool.html>. Il programma deve innanzitutto sapere quali sono le partizioni Reiserfs. Se ce n'è una sola, è sufficiente lanciare il programma senza alcuna opzione. Questo ne scriverà l'indirizzo nel Registro di sistema di Windows alla chiave `HKEY_CURRENT_USER\Software\p-nand-q.com\rfstool`. Se invece le partizioni sono più d'una, è possibile specificare quella da leggere sotto forma di variabile d'ambiente usando in una finestra DOS un comando come: `c:\> set REISERFS_PARTITION=1.0`, dove il primo numero indica il disco fisso e il secondo la partizione. `Hda1` è indicato con 0 e `hdb` con 1; allo stesso modo la prima partizione con 0, la seconda con 1 ecc. Il terzo metodo è sempre da finestra DOS, e ricorre all'opzione `-p`. Perciò il comando sarà: `c:> rfstool -p1.0`. Per copiare dei file il comando sarà invece del tipo: `c:> rfstool cp -p0.1 /root c:;`, che copia tutta la struttura e i file di `/root` in `c:`, e via dicendo.

19.15 Scrivere su NTFS

Abbiamo già rilevato come la possibilità offerta dal kernel 2.6 di scrivere su partizioni NTFS sia estremamente limitata. Esistono tuttavia alcuni strumenti per ovviare a questo problema.

Il primo è **captive**, il cui sito sta all'indirizzo: <http://www.jankratochvil.net/project/captive>. Esso tuttavia sembra abbandonato (l'ultima release risale al gennaio 2004) e comunque funziona, sebbene in maniera molto affidabile, solo con il kernel 2.4.

Il secondo è il progetto **ntfsprogs**, la cui homepage si trova all'indirizzo: <http://www.linux-ntfs.org>. Si tratta di una serie di utilità che consentono di lavorare sul file system NTFS in user-space, senza pertanto implicare in alcun modo le limitate possibilità dei driver nativi del kernel. È sufficiente montare con l'utilità `ntfsmount` la partizione di Windows, e quindi è possibile scriverci. Molte informazioni sul funzionamento di `ntfsprogs` si leggono (e in lingua italiana) all'indirizzo: <http://wiki.linux-ntfs.org/doku.php?id=ntfs-it>.

Esiste infine un prodotto commerciale, **mount everything**, informazioni sul quale si possono consultare all'indirizzo: <http://www.mount-everything.com>, che consente la lettura e la scrittura anche con il kernel 2.6.

Recentemente, è stato annunciato un driver open source battezzato **ntfs-3g** in grado di effettuare illimitatamente sia la lettura sia la scrittura su NTFS. Il comunicato si può leggere al seguente indirizzo: http://sourceforge.net/mailarchive/forum.php?thread_id=23836054&forum_id=2697. Il driver, ancora in fase di testing, è scaricabile all'indirizzo: <http://mlf.linux.rulez.org/mlf/ezaz/ntfs-3g-20070714-BETA.tgz>.

Ringraziamenti

Vorrei qui ringraziare Loris, il maintainer di `slacky.it`, e Giorgio (Useless), con la loro accurata conoscenza dei sistemi Unix, che hanno pazientemente letto versioni preparatorie di questo scritto a caccia dei miei errori, suggerendomi importanti correzioni, integrazioni e modifiche.

Ringrazio inoltre Paoletta per i suggerimenti sul capitolo dedicato a connettività e reti, Christian Eric (Chrix), Liquid e Firetux per i rilievi sul capitolo dedicato a Samba.

Infine, ringrazio Luigi Carusillo, autore fra l'altro di alcune significative guide su \LaTeX , che mi ha aiutato nella preparazione del codice di questo documento.

Resta inteso che la responsabilità finale di *Slackware for dummies* è soltanto mia. Ogni segnalazione, consiglio o richiesta sono sempre ben accetti. L'Autore è contattabile al seguente indirizzo e-mail: `mauro.sacchetto@alice.it`.

Dedico questa guida alla mia famiglia: a mia moglie Sabina, che spesso è andata a dormire da sola mentre io continuavo a dedicarmi a Slackware, a mia figlia Francesca, che mi ha spento per gioco il computer, facendomi perdere tutto il lavoro fatto, solo due o tre volte, e al piccolo Luca, arrivato da pochi mesi.

The GNU General Public License

Version 2, June 1991
Copyright © 1989, 1991 Free Software Foundation, Inc.
51 Franklin St, Fifth Floor, Boston, MA 02110-1301, USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Abstract

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software – to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation’s software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead). You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author’s protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors’ reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone’s free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU General Public License

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The “Program”, below, refers to any such program or work, and a “work based on the Program” means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term “modification”). Each licensee is addressed as “you”.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program’s source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:
 - (a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.
 - (b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.
 - (c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement).

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License,

whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:
 - (a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
 - (b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
 - (c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.
5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.
7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.
9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.
Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.
10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.
12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

Appendix: How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

```
<one line to give the program's name and a brief idea of what it does.>  
Copyright (C) <year> <name of author>
```

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA 02110-1301, USA.

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

```
Gnomovision version 69, Copyright (C) <year> <name of author>  
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type 'show w'.  
This is free software, and you are welcome to redistribute it under certain conditions; type  
'show c' for details.
```

The hypothetical commands `show w` and `show c` should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w` and `show c`; they could even be mouse-clicks or menu items – whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a “copyright disclaimer” for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program
'Gnomovision' (which makes passes at compilers) written by James Hacker.

<signature of Ty Coon>, 1 April 1989
Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.