

for more please visit :

<http://articlopedia.gigcities.com>

SLACKWARE FOR DUMMIES

ovvero

Come un inesperto può usare Slackware 10.0

di Mauro Sacchetto

1° ed: giugno 2004

2° ed: luglio 2004



Indice

1. Introduzione

2. Procurarsi Slackware

3. Installare Slackware da CD

3.1. Partizionare il disco fisso

Scheda: Che cos'è la partizione /home

3.2. Installare Slackware

Scheda: I kernel di Slackware 10.0

Scheda: I pacchetti di Slackware 10.0

3.3 Installare Slackware da floppy

Scheda: Le directory di Slackware 10.0

4. Avviare Slackware per la prima volta

4.1. Ottimizzare il server grafico

4.2. Ottimizzare il server sonoro

Scheda: Che cosa sono i permessi

5. Rifinire la configurazione di Slackware

5.1. Gestire gli utenti

5.2. Gestire i gruppi

5.3. Montare nuove unità

Scheda: Il file /etc/fstab

5.4. Installare e gestire pacchetti in Slackware

Scheda: I comandi di pkgtool

5.5. Creare un multiboot

Scheda: Il file /etc/lilo.conf

6. Installare periferiche

6.1. Installare una stampante

6.2. Installare uno scanner

6.3. Installare una webcam

7. Condividere file con NFS

8. Tips & trick

- 8.1. Velocizzare l'avvio di Slackware**
- 8.2. Modificare il manager di avvio**
- 8.3. Avviare Slackware in modalità grafica**

Scheda: Inizializzazione e runlevel

- 8.4. Lanciare programmi all'avvio**
- 8.5. Spegner Slackware**
- 8.6. Italianizzare Slackware**
- 8.7. Tradurre le man pages in italiano**
- 8.8. Accendere il tastierino numerico all'avvio**
- 8.9. Installare fonts**
- 8.10. Modificare i font della console**
- 8.11. Colorare la console**
- 8.12. Stampare con OpenOffice o con StarOffice**
- 8.13. Abilitare java in Mozilla**
- 8.14. Aprire link in Mozilla**
- 8.15. Rimuovere fortune**
- 8.16. Eliminare il messaggio "mysql ended"**
- 8.17. Installare il kernel 2.6.7**
- 8.18. Installare i driver per l'accelerazione grafica (schede ATI)**
- 8.19. Aggiornare Slackware con Swaret**
- 8.20. Impostare un firewall**

1. Introduzione

Da qualche tempo la distro GNU/Slackware Linux (nata nel 1993) sta prendendo piede, e – soprattutto grazie alle innovazioni apportate a partire dalla versione 9.0 – è divenuta maggiormente accessibile. Parecchi sono infatti gli strumenti di configurazione (da **xf86config** ad **alsaconf**, da **netconfig** a **ifconfig**, da **wxmconfig** a **ldconfig** 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 provengano da Windows (ma anche da altre distribuzioni Linux). Tuttavia ha un suo contraltare nient’affatto 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, ne rende assolutamente praticabile la ricompilazione, che in altre distribuzioni invece è una faccenda estremamente complicata. È poi facile accorgersi che, in molti casi, proprio questa linearità di Slackware consente di configurare il sistema con pochi ritocchi ai file di sistema. Inoltre Slackware, spogliata delle GUI (acronimo di Graphical User Interface, e cioè dei frontend grafici), gira perfettamente anche su vecchi computer dotati di scarse risorse.

Questa guida intende essere un breve ausilio per muovere i primi passi nella configurazione di Slackware, redatta in modo piano da un neofita e destinata ai neofiti che devono essere guidati “passo dopo passo”. Una guida esaustiva è impossibile da scrivere, perché la configurazione varia a seconda dell’hardware e delle esigenze dell’utente. 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. Una configurazione completa ed efficiente, per chi conosce Slackware, porta via ben poco tempo!

2. Procurarsi Slackware

Il primo sito in cui è possibile scaricare le ISO di Slackware è, naturalmente quello stesso della distribuzione, all’indirizzo <http://www.slackware.com>, che è tuttavia assai lento. Ci sono svariati mirror più veloci, facilmente individuabili con una ricerca su Google; alcuni sono reperibili anche in <http://www.slackware.com/getslack>. Come quasi tutte le distribuzioni Linux, anche Slackware si trova in allegato a parecchie riviste. Se si intende utilizzarla stabilmente, non è sicuramente inopportuno acquistarla o abbonarsi (per ricevere gli aggiornamenti), per fornire un supporto anche se minimo agli sviluppatori. Il set di CD acquistabili consta di quattro supporti perché ha anche tutti i sorgenti. Inoltre, nel CD 3, è presente anche lo Slackware-Book, ossia la guida ufficiale di Slackware.

3. Installare Slackware da CD

3.1. Partizionare il disco fisso

3.1.1. L’esigenza minimale di Slackware è avere due partizioni, una per l’installazione del programma e una per il file di swap (utile a reperire memoria nel caso sia insufficiente quella fornita dalla ram del computer); sarà questo il caso che considereremo. Naturalmente questa configurazione può essere modificata

a seconda delle esigenze di ciascuno. Ad esempio, la cartella /home di ciascun utente può essere creata in una partizione diversa. Nel caso si debba reinstallare il programma, ciò 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 Linux come dei file; ciò rende molto semplice costruire una serie di partizioni differenti in più dischi fissi.

Partiamo da due ipotesi: 1) un disco fisso che contiene già Windows (poiché nella maggior parte dei casi lo abbiamo già come sistema operativo o avendolo trovato preinstallato nel computer o avendolo installato noi); 2) un disco fisso nuovo.

3.1.2. 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 Linux) 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, oppure con uno strumento opensource come Parted (presente nella directory /extra del CD 3 di Slackware). Nel primo caso utilizzeremo 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 Linux di due formati, EXT2 ed EXT3, mentre non supporta altri formati come Reiserfs) e una di swap. La prima dev'essere sufficientemente ampia da ospitare il sistema operativo (quasi 3 GB) e i programmi aggiunti (un GB in più potrebbe essere sufficiente); la seconda, alla luce di un antico adagio, dovrebbe ammontare al doppio della ram disponibile. Tuttavia ciò è valido se ha poca ram; oggi, non essendo infrequente trovare computer domestici che hanno anche 1 GB di ram, si può creare un file di swap più piccolo. Se poi vogliamo, creeremo anche una partizione /home per l'utente principale (ciò vale anche nel caso che siamo noi l'unico utente del nostro computer). Anche in questo caso 1 GB può andare più che bene, ma poi tutto dipende da che cosa vogliamo metterci dentro. Se contiamo di scaricarci una enorme quantità di immagini o di file multimediali, basta aumentare a piacimento la dimensione.

3.1.3. Se vogliamo, possiamo creare la partizione direttamente 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.

3.1.3.1. In quest'ultimo caso dobbiamo preparare il disco con due passaggi. Innanzitutto creiamo le partizioni per i vari filesystem che intendiamo adottare usando **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). Lanciato il programma, se il disco fisso è vuoto il programma ci chiede se intendiamo attivarlo da "zero table"; rispondiamo affermativamente. Dal menu in basso, scegliamo Nuova partizione, primaria, avviabile, quindi digitiamone la dimensione e infine scegliamo il tipo di formattazione. Per Linux si tratta del filesystem 83, per il file di swap del filesystem 82. Ripetiamo i vari passaggi per ogni ulteriore partizione, naturalmente senza indicare queste nuove partizioni come avviabili. Quindi diamo l'ordine di scrivere, confermiamo con **si** (senza accento); una volta che il programma abbia terminato la scrittura, usciamo e riavviamo il sistema. A questo punto il co-

mando **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 **cmdisk**.

3.1.3.2. Ora dobbiamo formattare le partizioni create. Ci sono comandi diversi a seconda del filesystem prescelto:

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 /dev/hda1 ecc.).

mkswap /dev/nome_dispositivo = crea una partizione di swap. Prescindiamo qui dal fatto che, invece di una partizione, sarebbe possibile creare lo swap come un file (operazione che richiede qualche passaggio in più). Una volta creata, tale partizione dev'essere attivata col comando **swapon /dev/nome_dispositivo**.

3.1.3.3. Aggiungiamo, a titolo informativo, che le partizioni DOS si formattano col comando **mkdosfs**. Ad esempio, una partizione FAT32 si crea col comando **mkdosfs -F32 /dev/nome_dispositivo**.

Per sicurezza, dopo tutte queste operazioni è opportuno riavviare il sistema.

3.1.4. Nel caso si debbano trasferire dati da una partizione Windows NTFS a una Linux, è necessario creare un'ulteriore partizione "di scambio" in FAT32. Linux infatti legge e scrive regolarmente su FAT32, mentre è in grado di leggere soltanto il filesystem NTFS. Col kernel 2.6.x 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.

NOTA Non analizziamo qui né la procedura per la ricompilazione del kernel, né quella per la compilazione del nuovo kernel 2.6.x. 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>.

3.1.5. 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 apparirà 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 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 hdb 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 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; tuttavia, in genere, i dischi fissi sono collegati al primo canale e dispositivi come lettore, masterizzatore e unità DVD al secondo.

Che cos'è la partizione /home?

Linux è un sistema multiutente: ciò significa che prevede l'accesso allo stesso computer da parte di utenti diversi, che non devono poter accedere ai dati altrui. L'utente root 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, installare e personalizzare i programmi che usa senza che queste modifiche si rispecchino nelle varie home degli altri utenti.

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 normalmente come utente, è più difficile danneggiare il sistema stesso: 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 Linux) non possono danneggiare il sistema, perché non hanno accesso alle sue parti vitali.

Per la creazione di utenti non root, vedi sotto al n. **5.1**.

3.2. Installare Slackware

3.2.1. Acceso il computer e inserito il CD 1 (che contiene tutti i file essenziali al 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: ce ne sono numerosi, per rispondere a svariate esigenze di configurazione hardware (l'elenco completo dell'hardware supportato si legge nel file README.TXT presente nel CD 1, nella directory /bootdisk). 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.

I kernel di Slackware 10.0

Slackware 10.0 ci mette a disposizione i seguenti kernel:

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 CD-ROM/DVD.

bareacpi.i: kernel come bare.i, con l'aggiunta del supporto per ACPI (Advan-

ced Configuration and Power Interface). È così in grado di svolgere funzioni come l'APM (Advanced Power Management). Solitamente, si usa sui laptop.

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

lowmem.i: kernel utile per sistemi IDE con scarsa RAM (meno di 8 MB), che inoltre supporta i processori 386.

old_cd.i: kernel analogo al bare.i, con l'aggiunta di un supporto per una serie di vecchi lettori DC-ROM.

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

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

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

jfs.s: kernel come il bare.i, ma col supporto per l'IBM's Journaled Filesystem 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.

xfs.s: versione estesa di bare.i, col supporto per il filesystem XFS (SGI).

Qui prendiamo in considerazione solo l'ipotesi di un hardware senza componenti particolari (come dischi o masterizzatori SCSI): in questo caso, si sceglie la modalità proposta di default, e cioè il kernel bare.i.

3.2.2. 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, selezioniamo `qwerty/it.map` e diamo l'**OK**. Poi confermiamo la scelta, sempre col tasto **1**.

3.2.3. Ora possiamo fare il login come root, avviando il setup vero e proprio al prompt (che appare: `root@slackware:/#`) digitando **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 **setup**.

3.2.4. Dobbiamo quindi scegliere le partizioni. Poiché il (primo) disco fisso è identificato come `hda` (sappiamo già che un eventuale secondo disco fisso risulterebbe `hdb`), scegliamo la partizione di swap (nel nostro esempio `hda3`). Se l'abbiamo già formattata come tale, Slackware la riconosce automaticamente e ci propone di riformattarla con o senza il controllo dell'integrità (che non fa mai male) 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.

3.2.5. Scegliamo quindi la partizione su cui installare il sistema operativo (nel nostro esempio `hda2`), e battiamo invio. Anche in questo caso, se l'abbiamo già formattata essa verrà riconosciuta subito, ma possiamo effettuare un'ulteriore scelta. Possiamo cioè mantenere il filesystem **EXT3** oppure riformattarla **EXT2** o **Reiserfs**. **EXT2** è il classico formato Linux, **EXT3** è la versione journalized di

EXT2, **Reiser** è un filesystem più nuovo, ma molto veloce, che organizza le informazioni in una specie di database per cui è in grado di disporre e trovare anche un grandissimo numero di file di piccole dimensioni in tempi molto rapidi. I filesystem nativi di Linux hanno un'ottima gestione, e conseguentemente non necessitano di essere deframmentati, come accade invece nei filesystem di Windows. Slackware propone Reiserfs come default. Anche in questo caso il programma di installazione può effettuare una formattazione veloce oppure con verifica (che in realtà è anch'essa molto veloce). Una volta che la partizione selezionata è stata vista dal programma come "in use", confermiamo che abbiamo terminato di aggiungere partizioni ("Done adding partitions") e facciamo aggiungere le informazioni in /etc/fstab.

3.2.6. 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 in FAT32). In questo caso dobbiamo indicare un punto di montaggio per potervi accedere in seguito (un "pick mount pointer"): lo indicheremo con un nome a piacere, precisando che, secondo la migliore tradizione, va montata in /mnt (scriviamo cioè una cosa del tipo /mnt/nome_della_partizione).

3.2.7. Infine dobbiamo scegliere i programmi. Slackware continua a utilizzare una distribuzione dei programmi in pacchetti indicati da lettere dell'alfabeto o da brevi sigle: la cosa ha origine nel fatto che le sue prime versioni risalgono ad anni in cui non erano disponibili i CD-ROM, per cui i programmi stessi erano distribuiti in una molteplicità di floppy.

I pacchetti di Slackware 10.0

Vediamo in breve che cosa contengono i vari pacchetti presenti in Slackware 10.0 (un descrizione più dettagliata si legge nel file PACKAGES.TXT presente nel CD 1).

Pacchetti della directory /slackware del CD 1

Pacchetto serie a: 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.

Pacchetto serie ap: 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.

Pacchetto serie d: contiene programmi di sviluppo, come compilatori per i sorgenti (autoconfs, automake), cvs, interpreti di vari linguaggi come gcc (per C e C++), Perl, Python ecc. Essenziale agli sviluppatori, serve comunque per gestire Slackware (senza di esso, ad esempio, non saremmo in grado di compilare alcun pacchetto dai sorgenti).

Pacchetto serie e: contiene l'editor emacs, che presenta però molte altre potenzialità.

Pacchetto serie f: contiene il materiale informativo, e cioè le FAQ (Frequently Asked Questions) e gli HOW-TO.

Pacchetto serie k: contiene i sorgenti del kernel (2.4.26).

Pacchetto serie l: contiene le librerie di sistema. Servono ovviamente agli svi-

luppatori, ma sono essenziali per il funzionamento della maggior parte dei programmi. 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).

Pacchetto serie n: contiene gli strumenti di networking, cioè per la navigazione in rete. Dunque PPP (per la connessione a Internet), alcuni browser a console (lynx), svariati server per il web (Apache), per il servizio DNS (Bind), server e client per la posta elettronica (fetchmail, procmail, mutt, pime) 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.

Pacchetto serie t: contiene Tex, strumento per editare file.

Pacchetto serie tcl: contiene il supporto per il linguaggio di programmazione TCL/TR scripting language.

Pacchetto serie x: contiene i file essenziali allo X server, cioè il server grafico. È con questo linguaggio che è stato sviluppato X, appunto il server per la gestione degli windows manager.

Pacchetto serie xap: contiene le applicazioni che girano sotto il serve grafico. Pertanto include svariati windows manager (blackbox, fluxbox, fvwm, fvwm95, windowmaker), browser (Mozilla, Netscape), client per chattare (gaim, xchat), server news (pan), visualizzatori di immagini (imagemagic, xv), visualizzatori di file *.pdf (xpdf), programmi di grafica (Sane e Xsane per usare lo scanner), fotoritocco (gimp), videoscrittura (Xvim), strumenti multimediali (Xmms, Gxine), xscreensaver (raccolta di screensavers) e altro ancora.

Pacchetto serie y: contiene alcuni giochi.

Pacchetti della directory /slackware del CD 2

Pacchetti serie KDE: tutti i pacchetti che compongono KDE e cioè:

- 1) **kdeaccessibility:** utilità per facilitare la visione dello schermo e l'uso del mouse.
- 2) **kdeaddons:** plugin e script addizionali per alcune applicazioni di KDE.
- 3) **kdeadmin:** le utilità per la gestione di KDE.
- 4) **kdeartwork:** temi, screensaver, suoni, sfondi e finestre addizionali.
- 5) **kdebase:** i file base di KDE. Include kdm (lo windows manager di KDE/, konqueror, konsole, kaudio (il server audio di KDE), i file della guida e parecchie altre componenti.
- 6) **kdebindings:** librerie per i vari linguaggi.
- 7) **kdeedu:** programmi didattici di KDE.
- 8) **kdegames:** i giochi di KDE.
- 9) **kdegraphics:** i programmi di grafica di KDE. Include visualizzatori per vari formati e altri strumenti.
- 10) **kdelibs:** le librerie necessarie a KDE. Include anche arts, il demone sonoro di KDE.
- 11) **kdelinks:** strumenti di integrazione per KDE e Gnome.
- 12) **kdemultimedia:** programmi multimediali di KDE (lettori cd, mp3, mixer, ecc).
- 12) **kdenetwork:** strumenti per la navigazione in rete.
- 13) **kdepim:** il KDE Personal Information Manager: utilità per la gestione delle informazioni personali.
- 14) **kdesdk:** applicazioni per gli sviluppatori.
- 15) **kdetoys:** applicazioni minori di KDE.
- 16) **kdeutils:** utilità varie (visualizzatori di testo semplice, formattatori di flop-

py ecc.).

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

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

19) **qt**: strumento di sviluppo per sviluppare applicazioni con interfaccia grafica in C++.

20) **quanta**: compilatore HTML.

Pacchetti serie KDEI: i pacchetti col supporto linguistico per una novantina di lingue diverse.

Pacchetto serie GNOME: contiene tutti i pacchetti che compongono GNOME. Fra questi segnaliamo:

Abiword: programma molto flessibile di videoscrittura.

Bug-buddy: client IRC.

Control-center: strumento centralizzato per configurare GNOME.

Enlightenment: window manager basato su Mozilla.

File roller: gestore di archivi (compatta e scompatta).

Gconf: il database delle applicazioni di Gnome.

Galeon: il browser di default di Gnome, basato sullo stesso motore di rendering di Mozilla.

Gdm: il desktop manager di gnome desktop manager.

Glade: strumento per la creazione di interfacce grafiche con le librerie GTK+.

Gftp: client FTP.

Gnome-games: raccolta di giochi.

Gnome-media: pacchetto con innumerevoli strumenti multimediali.

Gnome-icq: programma per chattare.

Gnome2-users-doc: la guida di GNOME.

Gnumeric: foglio elettronico.

Gpdf: visualizzatore di file in formato pdf.

Gtk-engines: le librerie GTK, il linguaggio su cui si basa GNOME.

Nautilus: browser per la gestione delle risorse.

Nautilus cd burner: utilità per bruciare CD, integrato in Nautilus.

Totem: movie player basato su xine.

Pacchetti della directory /extra del CD 3

Negli "extra" Slackware ci offre una serie di programmi opzionali. Fra quelli che in realtà 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 compilare file *.tgz partendo dai sorgenti (su checkinstall vedi sotto al n. **5.4.6.**).

k3b: sontuoso frontend per la masterizzazione di CD e DVD.

parted: programma per la gestione delle partizioni.

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. Non installare, ad esempio, le librerie GTK richieste da Gnome perché non lo useremo ci impedirà anche di usare altri programmi sotto KDE che però richiedono quelle

stesse librerie. Pertanto, dei vari sistemi di installazione proposti, la soluzione gordiana è la prima, full installation, che installa tutto senza chiedere conferma. Tuttavia ci si troveranno così decine di KDE e di Koffice, include quelle in thailandese e in zulu, di cui pochi verosimilmente faranno uso. Una seconda opzione utile è l'installazione newbie, cioè per i principianti, che installa automaticamente i pacchetti essenziali e propone quelli opzionali, fornendone una breve descrizione. Ma in questo caso potremmo trovarci in difficoltà, perché in molti casi non sapremo se è il caso o meno di installare un certo programma. È allora meglio usare una terza opzione (scartando le rimanenti, destinate a utenti più esperti), quella detta menu. Essa effettua l'installazione selettiva per singoli pacchetti. In questo caso è conveniente selezionare tutto in tutti i pacchetti tranne KDEI, dove sceglieremo solo il supporto per la lingua italiana per KDE e per KOffice.

Inizia quindi la decompressione e la copia dei file, che a un certo punto ci chiederà di inserire anche il secondo CD (che contiene i pacchetti opzionali), e la configurazione del sistema.

Terminata questa fase, ci sono alcune altre cruciali operazioni da effettuare. Il programma di installazione ci chiede infatti una serie di cose.

3.2.8. 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.

3.2.9. Possiamo creare un **floppy di avvio**, cosa da fare assolutamente. Nel caso avessimo dei problemi a far partire il programma dal disco fisso, saremmo comunque sempre in grado di ricorrere al floppy. In realtà è sempre possibile far partire il sistema anche da CD, dal momento che il CD 1 e i CD2 sono in grado avviare il sistema, a patto che si specifichi la directory di boot (nel nostro caso, /dev/hda2).

3.2.10. Possiamo configurare il **modem**. Ovviamente, i passaggi ulteriori dipendono dal tipo di modem che possediamo e dalla porta in cui è inserito.

3.2.11. Possiamo abilitare **HOTPLUG** all'avvio. È una scelta da fare, perché hotplug si occupa di individuare e gestire le periferiche plug&play.

3.2.12. Ora dobbiamo installare **Lilo**, il bootloader (Lilo sta per 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 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.

È importante ricordare alcune cose: a) il bootloader di 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). b) Il bootloader di Windows viene così sovrascritto, ma se poi decidessimo di disinstallare Slackware, per eliminare anche Lilo e avviare Win-

dows sarà sufficiente ripartire con un floppy di avvio di Windows e lanciare il comando **fdisk /mbr**. Windows ripartirà senza problemi. c) Abbiamo qui supposto di avere già Windows nel sistema: se infatti si installa Windows dopo Slackware, il suo bootloader sovrascriverà Lilo. Di conseguenza sarà necessario entrare in Slackware col floppy di avvio e poi da lì reinstallare Lilo (vedi sotto al n. **5.5.6.**).

3.2.13. Possiamo configurare il **mouse**, scegliendo quello in nostro possesso. Slackware ci propone ben venti opzioni, a seconda del mouse che possediamo. Nel prosieguo considereremo solo il caso di un mouse PS/2 (il tipo ancor oggi maggiormente diffuso e che non crea problemi con Linux).

3.2.14. Possiamo abilitare **GPM**, anche in questo caso scelta conveniente. GPM rende possibile l'uso del mouse all'interno della console (la "finestra di comando" di Linux), cosicché potremo scorrere il testo con la rotellina del mouse o fare dei comodi copia / incolla altrimenti impossibili (per copiare il testo si preme la rotellina nel punto di inserimento).

3.2.15. 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. Il programma di installazione offre tre alternative: Static IP (se possediamo una scheda di rete a cui attribuire 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, ma si connette mediante un modem seriale. Qui la scelta varia ovviamente a seconda del nostro hardware.

Se ci troviamo nella prima situazione e abbiamo una connessione permanente, effettueremo la configurazione dapprima verificando l'indirizzo IP della scheda di rete col comando **ifconfig** (dovrebbe risultare fra l'altro un output del tipo 192.168.0.2). Se per qualche motivo dovesse essere necessario modificare il preesistente indirizzo IP della scheda, basterà lanciare il comando **ifconfig eth0** seguito dal nuovo indirizzo.

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.
```

Il mancato riconoscimento può essere dovuto a problemi hardware, ma anche alla mancata compilazione, nel kernel, del supporto necessario, nel qual caso si rende necessaria la sua ricompilazione).

Dobbiamo quindi lanciare il comando **netconfig** e inserire nel menu pseudo-grafico un HOSTNAME e un DOMAIN NAME a piacere (per ovvi motivi, è vivamente sconsigliato scegliere il nome di un dominio esistente, il che potrebbe produrre anche conseguenze legali), i quali identificheranno il computer, l'indirizzo IP (e cioè quello della nostra scheda di rete, qualcosa come 192.168.0.2), la netmask (di default per le reti di classe C è 255.255.255.0), il gateway (l'indirizzo del router, cioè 192.168.0.1) e il name server (ancora una volta 192.168.0.1). **Netconfig** scrive questi dati nel file **/etc/rc.d/rc.inet1.conf**.

In altri casi, tutto dipende dal tipo di modem e della connessione. Dato il grandissimo numero di variabili, un approccio produttivo si può avere cercando sui newsgroup e nei numerosi HOW-TO reperibili in rete.

3.2.16. 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** (su cui vedi sotto la scheda relativa). 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. Per un uso del servizio di risoluzione dei nomi, vedi sotto al n. **7.1.4.**

Cups = il principale gestore delle stampanti. Poiché supporta un vastissimo numero di stampanti ed è di facile configurazione, è meglio preferirlo a **lprng**. Va perciò attivato. Esso è raggiungibile da un browser all’indirizzo IP <http://localhost:631>.

Httpd = il web server Apache. È possibile accedervi aprendo un browser e digitando l’indirizzo IP <http://127.0.0.2>, dove è possibile rinvenire anche la documentazione. Questo servizio serve solo a coloro che vogliono fare del loro computer un server locale accessibile dall’esterno. In quest’ultimo 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 BDS Inet Ddemon, 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 esso 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_forwarding = serve se il computer deve fungere da router. Normalmente è un servizio che non viene utilizzato, se non per esigenze particolari.

Lprng = un gestore delle stampanti come cups. Visto abbiamo scelto quest’ultimo, non attiviamo lprng.

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 (se comunque si vuole rimuovere questo messaggio, vedi sotto al n. **8.16.**). Sarà un compito successivo imparare a usarlo.

Pcmcia = supporta le schede per i laptop.

Portmap = RPC portmapper demon. Serve a gestire NFS, ossia per usare il Network file system qualora sia nostra intenzione condividere file con altri computer aventi Linux come sistema operativo (vedi sotto al n. **7.**).

Samba = il gestore delle reti tra Linux e Windows. Serve a condividere file fra computer aventi Linux come sistema operativo e altri aventi Windows.

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.

3.2.17. 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.

3.2.18. Impostiamo adesso l'**ora di sistema**. Le alternative sono UTC e Ora locale. Scegliamo Local time, e andiamo con PgD a Europe, Rome. In ogni caso, UTC (acronimo di Tempo Universale Coordinato) può essere attivato sui computer che montano solo 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.

3.2.19. 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** (su cui, più in dettaglio, vedi sotto al n. **8.2.**). Slackware offre innumerevoli Window Manager:

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.

Gnome: si tratta l'altro Desktop environment più ricco e sviluppato, anche se leggermente più leggero di KDE. Ovviamente, la scelta è del tutto soggettiva.

XFCE: presenta un'interfaccia grafica leggera, ma funzionale.

Blackbox: presenta un'interfaccia leggerissima. Il desktop è vuoto e i programmi si aprono con menu a tendina attivati dal mouse.

Fluxbox: interfaccia grafica scarna, ma che offre la possibilità di aprire innumerevoli "linguette" per i vari programmi.

Wmaker: interfaccia grafica leggera, ma funzionale e dotata di innumerevoli strumenti. Inoltre è altamente personalizzabile.

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

FVWM95: versione più performante di FVWMZ. Presenta, in basso, un pannello dei programmi che ricorda un altro celebre sistema operativo...

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.

3.2.20. Ci viene richiesta una password di root. È opportuno scegliere, come sempre, una stringa non troppo semplice od ovvia. Se risulta troppo semplice (ad esempio breve e composta da sole lettere minuscole), 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).

3.2.21. A questo punto, non resta che aspettare il completamento dell'installa-

zione e quindi riavviare il sistema con **^+Alt+Canc**, dopo aver tolto il CD dal lettore.

3.3. 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` 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:

da `/bootdisks` lanciare il comando `cat bare.i > /dev/fd0` per creare il BOOT DISK,

da `/rootdisks` lanciare il comando `cat install1 > /dev/fd0` per creare il ROOT DISK 1,

da `/rootdisks` lanciare il comando `cat install2 > /dev/fd0` per creare il ROOT DISK 2.

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.

Le directory di Slackware 10.0

L'organizzazione di Slackware non è troppo dissimile da quella delle altre distribuzioni Linux, tuttavia non manca di alcune peculiarità. Vediamo com'è articolata la struttura delle principali directory.

/bin: contiene file binari (cioè eseguibili) essenziali per il funzionamento del sistema operativo, accessibili a tutti gli utenti.

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

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

/etc: contiene gli script e i file di configurazione 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 e (nella subdirectory `/modules`) i moduli del kernel.

/mnt: contiene i punti di montaggio delle periferiche.

/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. Contiene informazioni sul kernel e sui processi in esecuzione.

/root: la directory principale dell'utente root.

/sbin: contiene i comandi del sistema operativo, che solo root può lanciare. Se un utente cerca di lanciarli dal suo account, riceverà il messaggio: "command not found". Per trasformarsi in root e poterli così lanciare, si darà il comando `su` per restare comunque sempre nella home dell'utente o il comando `su - root` per trovarsi nella home di root.

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

/tmp: contiene i file temporanei, ma anche una serie
/usr: contiene i programmi a disposizione degli utenti. La subdirectory /usr/X11R6 contiene l'X system del sistema, i cui file di configurazione risiedono invece in /etc/X11.
/var: contiene i log di sistema e dei file gestiti dai processi di sistema.

4. Avviare Slackware per la prima volta

4.0.1. Dopo il riavvio e il caricamento del programma, ci si presenta il prompt, nella forma `root@darkstar:-#`. Dobbiamo inserire `root` come username e la password che abbiamo scelto.

4.0.2. 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`, alla ricerca di eventuali errori o altro. In alternativa, possiamo andare a leggere i file `/var/log/syslog` e `/var/log/messages`.

4.0.3. A questo punto, facciamo partire il nostro Desktop environment col comando `startx` (X è il nome del server grafico), e si caricherà KDE. Passiamo così alla prima, rapida configurazione di KDE, scegliendo lingua ed effetti.

4.1. Ottimizzare il server grafico

4.1.1. Per ottimizzare la resa dello schermo, dobbiamo tuttavia impostare alcuni elementi del server grafico. Mentre Slackware 9.1 ci offriva a tale proposito tre strumenti (`xf86config`, in modalità testuale, `xfree86setup`, in modalità pseudo-grafica, e `xf86cfg` in modalità grafica), con l'adozione di Xorg si può lanciare o `xorgconfig` in modalità testuale o `xorgsetup` in modalità pseudo-grafica. Scegliamo la prima possibilità: apriamo dunque una console e digitiamo `xorgconfig`. Ci si apre così un menu molto intuitivo, che però richiede una conoscenza minimale del nostro hardware.

4.1.2. Molte delle domande che il programma pone sono assolutamente intelligibili, e le risposte molto variabili a seconda del nostro hardware. Si inizia con la scelta del mouse: se abbiamo un mouse PS2 con rotellina, non abilitiamo l'emulazione per i 3 bottoni; la rotellina avrà bisogno di un piccolo, ulteriore intervento per funzionare (vedi sotto al n. **4.1.4.**). Accettiamo il dispositivo di default (`/dev/mouse`). Scegliamo quindi il tipo di tastiera, solitamente una generica a 105 tasti (o non funzioneranno i caratteri `<` e `>`) e la lingua (l'italiano è il numero 43). Soprassediamo alle ulteriori opzioni XKB, 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 la gestione dei led luminosi relativamente ai tasti.

Per settare il monitor, accettiamo la frequenza di refresh orizzontale o scegliamo 11 per impostare quella del nostro monitor; stessa operazione per la frequenza verticale. Diamo un nome a piacere che identifichi il monitor. Scegliamo la scheda video dal database, quindi la sua memoria, infine attribuiamo un nome a piacere che identifichi la scheda. Se i modi di risoluzione sono corretti, accettiamo. Specifichiamo la profondità di colore del monitor. Questi sono tutti dati di cui dovremmo già essere in possesso, e che sono comunque riportati nel manuale dello schermo stesso. Alla fine salviamo il file (di testo) contenente tutte le impostazioni, che è `/etc/X11/xorg.conf`.

4.1.3. 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 `vi` o `emacs`. La descrizione di questi programmi esula dall'argomento di questa trattazione. Tuttavia, a chi si dovesse trovare in difficoltà suggeriamo di aprire il file con `vi`, lanciando il comando **vi /etc/X11/xorg.conf**; è possibile quindi scorrere il testo e attivare la modalità di inserzione testo (ossia di scrittura) premendo il tasto **Esc** e quindi il tasto **i** per scrivere dal punto di inserzione oppure il tasto **a** per scrivere immediatamente dopo il punto di inserzione. Una volta modificato il testo scrivendolo normalmente sulla tastiera (per cancellare una lettera in modalità di comando premendo **Esc** e digitando quindi **x**), si esce salvando le modifiche premendo il tasto **Esc** più volte fino a sentire in segnale sonoro e quindi digitando **:wq** seguito da **Invio**. Se invece si volesse uscire senza salvare, bisognerebbe digitare **:q!** seguito da **Invio**.

4.1.4. 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 (il più semplice, in questo momento, è per noi `Kwrite`). Andiamo alla sezione "Input Device", e nella sottosezione "Identifier" "Mouse" modifichiamo la riga:

```
Option "Protocol" "PS/2"
```

in

```
Option "Protocol" "IMPS/2"
```

e quindi aggiungiamo sotto le righe:

```
Option "ZAxisMapping" "4 5"
```

```
Option "Emulate 3Buttons" "no"
```

4.1.5. Se è abilitata l'accelerazione grafica nel kernel dobbiamo decommentare (cioè togliere il cancelletto `#`), sempre in `/etc/X11/FX86Config` nella sezione "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.

4.2. Ottimizzare il server sonoro

4.2.1. Per ottimizzare la resa della scheda audio, innanzitutto abilitiamo da console il server sonoro, che di default potrebbe essere muto, col comando **alsamixer unmute**. Lanciamo quindi, sempre da console, il comando **alsamixer**. Ci si apre un menu pseudo-grafico in cui dobbiamo impostare i volumi delle varie applicazioni. 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` 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**.

Non resta che andare nel Centro di controllo di KDE, scegliere ALSA come dispositivo audio del sistema sonoro e fare gli ultimi aggiustamenti al mixer di

KDE, e cioè Aumix.

4.2.2. 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, dobbiamo 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.
```

4.2.3. In alternativa, è possibile ottenere lo stesso risultato in modo più lineare inserendo i nomi degli utenti nel gruppo audio. Ciò si può fare o aggiungendo manualmente i nomi nel file `/etc/group`, oppure da console col comando **usermod -G audio nome_utente**.

Che cosa sono i permessi

In 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. Più precisamente, lettura con **r** o con **4**, scrittura con **w** o con **2**, 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** (vedi sotto al n. **5.2**), 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: **rxwxrwxrwx**, e cioè **777**. Se solo root ha tutti i permessi e le altre due categorie solo il permesso di lettura, avremo: **rxwxr--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 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 e 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 "other", cioè tutti gli utenti non inclusi nel gruppo. Ad esempio, se volesse rendere eseguibile da tutti un file, root 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 i file, dovrebbe lanciare il comando **chmod o-r nome_file**; se infine volesse revocare a tutti utenti tranne che al proprietario il permesso di disporre in alcun modo del file, dovrebbe lanciare il comando **chmod o-rwx nome_file**.

È infine possibile cambiare il proprietario di un file col comando **chown** (che significa “change owner”), con la seguente sintassi:

chown nome_nuovo_proprietario nome_file.

Questi comandi supportano l’opzione **-R**, che modifica ricorsivamente permessi e proprietari anche delle subdirectory.

Quando si ha a che fare con sistemi diversi da Linux, che non presentano perciò in forma nativa i permessi (come accade con Windows), a volte 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 nella formula numerica 000, la negazione di qualsiasi permesso 777, mentre la concessione del permesso di lettura (e di esecuzione, per poter accedere ai file), ma non di scrittura al gruppo e agli utenti sarà 022 (complementare di **chmod** 755), e così via.

Ogni nuovo file ha un **umask** di default, che si può leggere in `/etc/profile` alla sezione Default umask, e che in Slackware è pari a 022. Ciò significa che gruppi e utenti hanno il permesso 5, e cioè di lettura ed esecuzione, ma non di scrittura. È possibile modificare questo settaggio editandolo manualmente.

La concessione di permessi speciali si può effettuare con **sudo** (acronimo di “superuser do”). Vi sono infatti determinate 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**. Qui non ci occupiamo in dettaglio di **sudo**; offriremo un solo caso delle sue funzionalità più sotto, al n. **8.5.3**.

5. Rifinire la configurazione di Slackware

È qui riportata la configurazione di alcuni altri elementi di Slackware, che reputiamo i più semplici e diffusi.

5.1. Gestire gli utenti

Linux è strutturalmente un sistema multi-utente. Abbiamo visto nella scheda **Che cos’è la partizione /home** i vantaggi di non lavorare da root, se non per compiti amministrativi.

5.1.1. La procedura di creazione di nuovi utenti è molto semplice, e naturalmente è un compito destinato esclusivamente a root. Da console, è sufficiente digitare il comando **adduser**, al che viene chiesto il **nome** del nuovo utente, il **numero** (uid = user id) da attribuirgli (i numeri uid degli utenti aggiunti iniziano da 1000, perché tutti i precedenti sono riservati al programma), e se non ci sono esigenze particolari va bene quello proposto di default, e cioè il primo numero libero, il **gruppo** fondamentale in cui collocarlo (di default è **users**), gli altri eventuali gruppi in cui collocarlo, la sua **home**, la sua **shell** (di default si tratta di bash), la **scadenza** (e cioè il tempo determinato per il quale quell’account funzionerà, dopo di che scade), altre eventuali informazioni, e infine la **password**.

L’utente ha accesso solo a una parte dei programmi, normalmente quelli conte-

nuti in `/bin`, `/opt`, `/usr`. Root può modificare i permessi 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 finger, e cioè le informazioni aggiuntive sull'utente (nome reale completo, indirizzo, numero telefonico, ecc.).

5.1.2. La rimozione di un utente è effettuata da root col comando **userdel**. Esso modifica anche 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; vengono infine eliminate la cartella `/home/utente` e il file dello spool della sua in mail: `/var/spool/mail/utente`.

Il comando **userdel -r** rimuove anche la home dell'utente.

5.1.3. In alternativa è possibile, sempre da parte di root, disabilitare un utente mantenendone però tutte le impostazioni. Ciò si effettua col comando **passwd -1** (l'opzione `-1` disabilita un account cambiando la password in un valore criptato che non coincide con nessun altro, mentre il comando **passwd -u** serve per riabilitare l'utente, ripristinando la password iniziale).

5.2. Gestire i gruppi

5.2.1. 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`.

5.2.2. Si crea un nuovo gruppo col comando **groupadd**. Si elimina un gruppo col comando **groupdel**. 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**.

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. 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 es.:

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 effettua modificando il proprietario con il comando **chown** oppure attribuendo a un gruppo creato apposta la possibilità di accedervi.

5.3. Montare nuove unità

5.3.1. Per aggiungere unità come partizioni del disco rigido, masterizzatori o pendrive, che Slackware non riconosce in automatico, è necessario suggerire a `/etc/fstab` la loro esistenza e alcune loro caratteristiche.

5.3.2. In 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, ai blocchi del quale si può accedere con un'operazione di **seek()**. Nei dispositivi a caratteri è invece possibile compiere operazioni di I/O relativi a un numero arbitrario di byte.

Dispositivi a caratteri:

le consoles = /dev/tty[n]

le porte seriali = /dev/ttyS[n]

Dispositivi a blocchi:

dischi rigidi = /dev/hda, /dev/hdb

cd-rom = /dev/hdc, dev/hdd

floppy = /dev/fd0

e inoltre masterizzatori, pendrive, dischi zip ecc.

5.3.3. Eventuali nuovi dispositivi possono essere creati in /dev col comando **mknod**, con la seguente sintassi:

mknod-m 660/dev nome_dispositivo b | c | u n_primario n_secondario
dove 660 o altro sono i permessi, *nome_dispositivo* può essere ad es. /dev/cdrom o /dev/sda1 (che in genere identifica dischi rimovibili o pendrive), **b** indica un dispositivo a blocchi, **c** indica un dispositivo a caratteri e **u** indica un dispositivo senza buffer. I numeri primari e secondari indicano il tipo di dispositivo e vengono indicati nel file /usr/src/linux/Documentation/devices.txt.

5.3.4. Il file /etc/fstab, caricato 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 il comando è **umount /dev/nome_dispositivo** oppure **umount nome_punto_di_montaggio**). È inoltre necessario creare nella directory /mnt dei punti di mount (che poi 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.

Il file /etc/fstab

Ogni riga del file è dedicata a un determinato dispositivo, secondo una sintassi del tipo:

dispositivo *punto di montaggio* *tipo di filesystem* *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 *hdxn*, dove *x* indica il disco e *n* la partizione. Così, *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 se-

conda partizione del secondo disco rigido.

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`: infatti `/dev/cdrom` non è che un link simbolico a `/dev/hdc`).

sd = disco fisso SCSI. Come per i dischi IDE, si avrà una successiva lettera e dei numeri per indicarne la partizioni. 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.

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 in 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).

Il filesystem

Linux supporta una enorme quantità di filesystem, anche vecchi o inconsueti. Qui sono menzionati solo i più diffusi. I valori che può assumere questa opzione sono:

auto = il sistema cerca di determinare automaticamente di quale tipo di filesystem si tratta. Ciò è utile nel caso di floppy o di pendrive, che possono essere formattati di volta in volta in diverso modo, e questo ci esime dal modificare ogni volta `/etc/fstab`.

ext2 = si tratta del filesystem nativo di Linux.

ext3 = si tratta di una versione aggiornata di ext2, che presenta il vantaggio di essere "journalled". Questo significa che viene tenuta traccia delle modifiche al filesystem, il che rende più semplice e veloce l'eventuale recupero dei dati in caso di crash del sistema.

reiserfs = si tratta di un filesystem particolare, che immagazzina l'elenco i dati in una specie di database. È molto veloce, adatto soprattutto se si devono manipolare file piccoli in grandissima quantità. Presenta lo svantaggio che non consente la transizione a ext2 o ext3.

iso9660 = si tratta del file system più diffuso dei CD-ROM.

msdos = si tratta del vecchio filesystem FAT16 di MS-DOS.

vfat = si tratta del filesystem FAT32 di Microsoft Windows 95/98/Me.

ntfs = si tratta del più recente filesystem elaborato da Microsoft per di Windows NT e WindowsXP.

Le opzioni di montaggio

Le più comuni e utili (per un elenco completo, **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 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, 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 agli utenti non root di montare il dispositivo.

nouser = 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.

exec = consente l'esecuzione di file binari (il contrario è **noexec**).

L'opzione dump

Questo valore è usato dal comando dump per effettuare dei backup sui filesystem. 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 per i filesystem non removibili.

L'opzione check

Questo valore è usato dal comando check per determinare l'ordine del controllo dell'integrità del filesystem 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 filesystem presenti nello stesso disco sono verificati in sequenza, mentre è possibile abilitare il check contemporaneo di filesystem 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.

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".

5.3.5. Per **montare partizioni del disco rigido o un altro disco rigido**, è necessario editare /etc/fstab aggiungendo, 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 mount, e cioè /mnt/windows e /mnt/archivio (i nomi delle directory in /mnt sono a piacere).

Nel primo caso abbiamo indicato a) la partizione del disco fisso che contiene il programma, b) la directory di mount, c) il tipo di filesystem, d) abbiamo dato indicazione di montarla automaticamente (cioè ogni volta all'avvio), abbiamo abilitato il montaggio di default ma abbiamo modificato i permessi con umask in modo che gruppo e utenti possano accedere all'unità; abbiamo poi specificato che si tratta di una unità in sola lettura. e) Abbiamo escluso il backup e il con-

trollo del filesystem.

Nel secondo caso con `gid` e `umask` combinati abbiamo autorizzato il gruppo 104 a effettuare tutte le operazioni e gli altri utenti ad accedere in sola lettura.

5.3.6. Per **montare un lettore CD e un masterizzatore**, è necessario creare un punto di mount in `/mnt`, con un nome a piacere, qui ad esempio `/mnt/cdrom` per il lettore e `/mnt/cdrw` per il masterizzatore. 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/cdrw iso9660 noauto,user,ro 0 0
/dev/hdd /mnt/cdrom iso9660 noauto,user,ro 0 0
```

In questo caso il filesystem è `iso9660`, il formato standard per i CD.

Un'ultima cautela per usare il masterizzatore: col kernel 2.4.x e con le vecchie versioni di `cdrecord` (il programma che serve a scrivere i CD) è necessario abilitare l'emulazione SCSI. È sufficiente aggiungere alla fine di `/etc/lilo.conf`, con un editor qualsiasi di testo, la riga:

```
append="/dev/hdc=ide-scsi /dev/hdd=ide-scsi".
```

In questo caso, lettore e masterizzatore non saranno più visti come dischi IDE, ma appunto come dischi SCSI, per cui non saranno indicati più come `/dev/hdc` e `/dev/hdd`, bensì come `/dev/scd0` e `/dev/scd1`.

Col kernel 2.6.x è possibile abilitare l'emulazione, ma la versioni più recenti di `cdrecord` non lo richiedono più lavorando tranquillamente coi masterizzatori IDE, per cui non serve aggiungere nulla in `/etc/lilo.conf`.

5.3.7. Per **montare un pendrive USB**, è necessario è necessario creare un punto di mount in `/mnt`, con un nome a piacere, ad esempio `/mnt/pendrive`. Quindi editare `/etc/fstab` aggiungendo la seguente riga:

```
/dev/sda1 /mnt/pendrive auto noauto,user,icharset=iso8859-15 0 0
```

Se poi compileremo il kernel 2.6.x, non tralasciamo di abilitare il supporto per i dischi SCSI e per l'hotplug, dal momento che questi dispositivi (e altri supporti come i dischi Zip) vengono riconosciuti per l'appunto come dischi SCSI.

Se si verificassero dei problemi, è innanzitutto opportuno verificare con che nome (device) viene riconosciuto il dispositivo. Lo si può fare inserendo il pendrive nella presa USB e verificando immediatamente l'output del comando **tail -F /var/log/messages**. È anche possibile abilitare gli utenti al montaggio aggiungendone i nomi nel file `/etc/group` alla voce "disk".

5.3.8. Per montare tutti i dispositivi elencati nel file `/etc/fstab` si deve lanciare il comandi **mount -a**. Per montare un dispositivo elencato in `/etc/fstab`, ma non caricato automaticamente, basta lanciare il comando **mount** seguito dal punto di montaggio (ad es. **mount /mnt/floppy**) e per smontarlo il comando **umount** seguito sempre dal punto di montaggio (ad es. **umount /mnt/floppy**).

5.3.9. 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 filesystem preceduto dall'opzione **-t**, c) il suo punto di montaggio. Ad esempio, per montare un floppy formattato FAT32 e collocato nella prima unità floppy: **mount /dev/fd0 -t vfat /mnt/floppy**; quindi, per smontarlo, **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.

5.4. Installare e gestire pacchetti in Slackware

5.4.1. Il sistema di gestione dei pacchetti di Slackware utilizza tar files compressi in modo comune (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.

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, da floppy, di rimuovere un pacchetto, di visualizzare il contenuto di un pacchetto (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 individuare nuovi fonts per il server grafico.

06. scrollkeeper: aggiorna il database della documentazione di Gnome.

07. get-register: lancia il registro gst (per i plugin gstreamer).

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 (di essi abbiamo parlato sopra, al n. **3.2.16.**).

setconsolefont: cambia i font della shell.

timeconfig: seleziona il fuso orario.

xwmconfig: seleziona lo windows manager di default.

5.4.2. Il comando **installpkg** serve a installare un pacchetto. È sufficiente dare 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`.

5.4.3. Il comando **upgradepkg** aggiorna un pacchetto che è già installato con il nuovo pacchetto specificato. È possibile fornire solo il nome del pacchetto nuovo col comando **upgradepkg nome_nuovo_pacchetto**, nel caso i due pacchetti abbiano lo stesso nome, ma sia diverso il numero di versione, mentre se hanno nomi diversi è necessario indicarli entrambi, pertanto col comando **upgradepkg nome_vecchio_pacchetto%nome_nuovo_pacchetto**, senza aggiungere alcuno spazio tra i nomi dei due pacchetti.

5.4.4. Il comando **removepkg** serve a rimuovere il pacchetto con il nome specificato. Pertanto è sufficiente dare il comando **removepkg nome_pacchetto**. 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`, dove `nome_pacchetto` 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.

5.4.5. Nel caso non siano disponibili i pacchetti *.tgz, ci sono svariate possibilità. Se abbiamo il pacchetto sorgente, solitamente in formato *.tar.gz (un archivio compresso: l'estensione -tar ci suggerisce che contiene innumerevoli file, mentre l'estensione -gz ci suggerisce che si tratta di un file compresso) oppure -bz2 (un formato di compressione migliore rispetto a -gz), possiamo compilare il programma. Solitamente la prassi consiste nel lanciare in sequenza i comandi **./configure**, **make** e infine **make install** (è comunque opportuno leggere sempre le note di installazione, perché a volte il procedimento differisce; ad esempio, alcune pacchetti recenti non richiedono più **./configure**). Eventualmente, a processo ultimato, si possono cancellare i file temporanei creati dall'installazione col comando **make clean**. Prudenza vuole che i primi due debbano essere lanciati da semplice utente e solo il terzo da root. In questo caso tuttavia il database non terrà conto del programma installato e sarà più difficile rimuoverlo; se il programma include la possibilità di disinstallazione, allora è possibile cancellarlo col comando **make uninstall**, ma per poterlo fare è necessario aver conservato i sorgenti sul disco fisso. Se infine desideriamo installare il pacchetto in una directory a nostra scelta, che non sia quella di default, è necessario dare il comando: **# configure -- prefix=/dir/dir**.

5.4.6. Partendo dai sorgenti, è possibile creare pacchetti in formato *.tgz. Anche in questo caso l'operazione si può effettuare in vari modi. Tralasciando qui di spiegare la compilazione manuale, operazione un po' complessa ma non estremamente difficile (a tale proposito si possono vedere due HOW-TO reperibili agli indirizzi:

<http://www.slack.it/misto/tutorial/spunleashed.txt>

e

<http://www.slack.z00.it/slack/slack.html>),

rileviamo come esista un utilissimo strumento per compiere questa operazione in modo automatico: **checkinstall**. Si tratta di un programma che Slackware non installa di default, ma che è reperibile negli extra, nel CD 3 della distribu-

zione (o scaricabile all'indirizzo: <http://asic-linux.com.mx/~izto/checkinstall>). Dai sorgenti, si deve dare prima `./configure` e poi `make`, come spiegato al n. **5.4.5.**, e quindi, invece di `make install`, 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 (i386, i486 ecc. Il Pentium 4 corrisponde a i686). In breve tempo avremo il nostro *.tgz pronto per l'installazione con `installpkg`.

5.4.7. È anche 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 pacchetto *.tgz porta con sé tutta la serie delle sue dipendenze (esattamente come l'rpm), che complicheranno fatalmente l'installazione.

5.4.8. L'elenco di tutti i pacchetti installati sta in `/var/log/packages`, mentre eventuali script di post-installazione in `/var/log/scripts/`. L'elenco dei pacchetti rimossi sta in `/var/log/removed_packages`, mentre quello di eventuali script di post-installazione in `/var/log/removed_scripts`.

5.5. Creare un dual boot

5.5.1. Molti utenti di 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 Linux. Per accedere a entrambi è necessario riconfigurare il bootloader di Slackware, e cioè Lilo. La prima possibilità che ci offre Slackware è configurarlo mediante il menu pseudo-grafico `lilo-config`. Tuttavia, per comprendere meglio il funzionamento di questo bootloader, scegliamo di configurarlo manualmente, con un editor di testo.

Il file `/etc/lilo.conf`

Il bootloader può essere installato sia nell'MBR (Master Boot Record), cioè la prima sezione del disco fisso, sia sulla partizione che contiene Linux. Mentre il bootloader di Windows non riconosce altri sistemi operativi (a meno di non modificare a mano `c:/boot.ini`), Lilo ci permette di scegliere agevolmente quale sistema lanciare all'avvio. Per fare questo dobbiamo aprire il file `/etc/lilo.conf` con un editor qualsiasi.

Facciamo innanzitutto attenzione alla sezione `# Start LILO global section`, che determina le opzioni globali. Qui ci interessano le righe:

`boot = /dev/hda`, che ci dice da dove viene effettuato il boot del sistema e dove installeremo Lilo;

`default = Slackware`, 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;

`timeout = 50`, che determina quanti secondi devono passare perché scompaia il prompt e si avvii il sistema di default, nel caso specifico 5 secondi (infatti il tempo è indicato in decimi di secondo).

Volendo, è anche possibile aggiungere la riga `password=password`, di modo che l'accesso sia limitato.

Consideriamo adesso la sezione che inizia: `# Linux bootable partition config begins`.

La riga `image = /boot/vmlinuz` indica il kernel di Linux che dev'essere caricato all'avvio e che si trova in `/boot`.

La riga `root = /dev/hdan` indica la partizione del disco fisso dove risiede la directory principale di Slackware (in questo caso, "root" non indica perciò l'utente root).

La riga `label = Slackware` indica semplicemente il nome del programma che incontreremo nel messaggio di boot.

Consideriamo infine la sezione che inizia `# DOS bootable partition config begins`.

La prima riga e cioè `other = /dev/hdan` indica la partizione in cui è situato il sistema operativo alternativo; la riga `label = Windows` indica ancora una volta il nome che intendiamo attribuire alla partizione (e che ritroveremo nel messaggio iniziale).

5.5.2. Per poter avviare anche Windows dovremo aggiungere una sezione del tipo:

```
# DOS bootable partition config begins
other = /dev/hdan
label = Windows
table = /dev/hda
# DOS bootable partition config ends.
```

5.5.3. 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.

5.5.4. 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 Linux, diciamo Mandrake, installata sulla quarta partizione del primo disco fisso (e dunque in `hda4`). Alla sezione `# Linux bootable partition config begins` e 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 Mandrake da caricare. Posto che da Slackware lo vediamo da `/mnt/mandrake`, la riga risulterà:

`/mnt/mandrake/boot/vmlinuz`. In b) indichiamo la partizione della directory root di Mandrake, nel nostro esempio `/dev/hda4`. In c) indichiamo il nome che ci apparirà nel messaggio di avvio, che potrebbe essere per l'appunto Mandrake.

5.5.5. In questo modo l'avvio di Mandrake è legato a Slackware, visto che è quest'ultima che gestisce tutti i programmi. Nel caso si cancelli Slackware, o si aggiorni Mandrake modificandone il kernel, si dovrà riconfigurare Lilo. Un'alternativa che eviti questa scomodità è ricorrere a un boot manager come GAG (Graphical Boot Manager), il quale consente multiboot e continua a risiedere nell'MBR del tutto indipendentemente dai sistemi operativi installati. Se adotta-

mo GAG, Lilo dovrà essere installato non nell'MBR, ma nel primo settore della partizione di root.

5.5.6. Nel caso, che talvolta si verifica, che durante l'installazione il programma non riesca a installare Lilo, è necessario avviare Slackware da floppy oppure da CD, e quindi lanciare in sequenza, come detto sopra **lilo -t** e **lilo -v**.

6. Installare periferiche

6.1. Installare una stampante

Se abbiamo settato Slackware in modo che all'avvio carichi automaticamente CUPS, e cioè il server di stampa, dobbiamo ora impostarlo. Apriamo da un browser il suo indirizzo IP, e cioè <http://localhost:631>, e scegliamo la voce Do Administration Task, quindi Printers, Add Printer e seguiamo la procedura che ci viene indicata. Naturalmente, non è detto che troveremo i driver per la nostra stampante, sebbene i modelli supportati siano ormai molto numerosi. Nell'eventualità, è necessario fare una ricerca in rete.

6.2. Installare uno scanner

6.2.1. Per installare uno scanner 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. È quindi necessario andare nella directory `/etc/sane`, individuare il file *marca_dello_scanner.conf* che corrisponde alla marca del nostro scanner (ed es. *epson.conf* oppure *hp.conf*), e lasciare decommentata solo la riga relativa al dispositivo suggeritoci da **sane-find-scanner**; nel caso di uno scanner USB, ad esempio, sarà `/dev/usb/scanner0` oppure `/dev/usbscanner0`. Infine dovremo controllare che sia decommentata la riga relativa alla marca dello scanner in `/etc/sane/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", possiamo usare tutti i programmi di scansione.

Per far sì che anche gli utenti possano usare lo scanner, è necessario modificare i permessi di `/dev/usb/scanner0` oppure di `/dev/usbscanner0` col comando **chmod a+x**.

6.2.2. Se è stato compilato il kernel 2.6, ci si trova davanti a una diversa gestione delle porte USB, che non dipende più da `/dev`, bensì da `libusb`. A proposito dell'installazione dello scanner in questo caso e del suo utilizzo sia da root che da user, cfr. il mio Mini-HOW-TO leggibile all'indirizzo: <http://www.slackware.it/tutorial/scanner/scanner-slackware.html>.

6.3. Installare una webcam

6.3.1. Il funzionamento della webcam USB implica che nel sistema siano presenti i driver corretti (e che sia attivato HOTPLUG). Il kernel di Linux supporta svariati modelli di webcam; in caso contrario bisognerà cercare in rete se ne esistono.

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 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 OV511+ 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+ Web-
Cam
Bus 002 Device 001: ID 0000:0000
Bus 001 Device 001: ID 0000:0000
```

dove quello che ci interessa non è né la stampante (Hewlett-Packard) né lo scanner (Epson), ma la terza voce.

In alcuni casi la webcam è riconosciuta come una generica periferica di acquisizione ottica, insomma più o meno come un 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.
```

6.3.2. L'installazione di Gnomemeeting è molto lineare, anche se richiede alcune librerie per poter funzionare. I pacchetti necessari sono i seguenti (le versioni sono quelle attualmente più aggiornate):

- a) gnomemeeting-1.0.2
- b)openh232-v_1_13_4
- c) openldap-2.2.11 (di cui in realtà servono solo le libldap)
- d) pwlber-v1_6_5

Gnomemeeting cerca quattro librerie che adesso esistono, ma con nomi diversi. Si devono quindi creare dei link simbolici in /usr/lib, coi seguenti comandi:

```
ln -s libldap-2.2.so.7.0.4 libldap.so.2
ln -s libldap_r-2.2.so.7.0.4 libldap_r.so.2
ln -s libldap_r-2.2.so.7.0.4 libldap-r.so.201
ln -s liblber-2.2.so.7.0.4 liblber.so.2
```

Lanciato la prima volta, Gnomemeeting con ogni probabilità accuserà una cattiva configurazione di Gconf (che in effetti Slackware gestisce in modo peculiare),

non riuscendo a trovare la chiave di registro `gconf_test_age`. Per rimediare, è 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.

7. Condividere file con NFS

7.1. 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 esclusivamente della condivisione di file in una rete interna (cioè in una LAN, acronimo di Local Area Network) fra computer che hanno tutti Linux come sistema operativo: in questo caso ci serviremo di NFS. C'è anche la possibilità di stabilire una condivisione fra computer che hanno Linux e altri che hanno Windows: in quest'altro caso ci si servirà 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 con **netconfig** (essi sono riscontrabili in `/etc/rc.d/rc.inet1.conf`).

7.2. Il protocollo NFS consente di "comunicare" a due o più computer che hanno un sistema operativo Linux. Slackware è già predisposta per questo servizio; possiamo comunque compiere una verifica col comando **vi /proc/filesystem**, 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 indirizzo IP dinamico stabilito da DHCP.

I demoni che (nelle versioni più recenti) 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 filesystem 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.

7.3. Configurazione del server

Nel file `/etc/exports` indicheremo a) la (o le) directory da condividere; b) il (o i) computer con cui condividerla (o condividerle), ossia il client; c) i permessi, racchiusi fra parentesi tonde e senza lasciare alcuno spazio con l'indicazione del client. a) si indica col percorso completo; la slash (/) indica che si intende montare l'intero filesystem. b) si indica con l'indirizzo IP oppure col nome del computer client (per quest'ultima possibilità, vedi al n. 7.5.). Si avrà pertanto una riga come:

```
/ 192.168.0.2(rw,nohide,_netdev,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,_netdev,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 filesystem 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 filesystem è disponibile in sola lettura (read only)

rw: il filesystem 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 di dati, ma può determinare corruzione dei dati.

nohide: se il server esporta due filesystem 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 filesystem 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 applicata lanciando il comando **exportfs -r**.

7.4. Configurazione del client

Si tratta adesso di montare la condivisione editando /etc/fstab, per indicare: a) l'indirizzo del server; b) la directory condivisa; c) il punto di montaggio; d) il filesystem, nel nostro caso sempre nfs; e) le opzioni. a) e b) sono scritte di seguito, separate solo dai due punti

Avremo perciò, ad esempio:

```
192.168.0.3:/ /mnt/brightstar nfs _netdev,sync,users 0 0
```

oppure

```
192.168.0.3:/home/samiel /mnt/brightstar nfs _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=1024: indica il buffer in scrittura (w = write). Attualmente il default è 4096, che può essere aumentato a 8192.

auto: monta automaticamente il filesystem.

noauto: non monta automaticamente il filesystem.

_netdev: indica che il filesystem 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 **inltr**.

inltr: 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.

7.5. 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.

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, a) l'indirizzo IP; b) il nome; c) 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)

7.6. Se il montaggio del filesystem 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 filesystem, 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**.

7.7. Per garantire un accesso controllato, i permessi possono essere definiti 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**.

8. Tips & tricks

Sono qui raccolte alcune delle tante procedure per settaggi particolari.

8.1. 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 controllare, col comando **/sbin/ldconfig**, se sono state installate nuove librerie. Normalmente ciò è inutile. Tale controllo si può 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
```

Sarà però necessario lanciare sempre **ldconfig** come root dopo aver installato nuove librerie o un nuovo programma.

8.2. Modificare il manager di avvio

Per modificare il manager di avvio 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.name`, dove `name` sta per il nome dei window manager installati, è sufficiente copiare quello prescelto in `/home` come `.xinitrc`.

8.3. Avviare Slackware in modalità grafica

Di default, il sistema non parte in modalità grafica. Dopo il caricamento, esso si blocca al prompt, col vantaggio che possiamo scegliere se entrare come root oppure come utente in modalità grafica, o ancora se desideriamo effettuare alcune operazioni 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).

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:
```

3 indica infatti 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: gdm (Gnome), kdm (KDE) e xdm (un display manager di base che consente soltanto di effettuare una registrazione nel sistema). Poiché **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 possibile, più semplicemente, commentare le righe relative a Gnome. Una terza soluzione, la più breve, 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.

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 BSD, laddove la più gran parte delle distribuzioni Linux ricorrono invece al System V. In breve, la differenza fondamentale consiste nel fatto che nel System V ogni runlevel ha una sua sub-directory con gli script di inizializzazione, mentre nello stile BSD è presente uno script per ogni runlevel.

Inizializzazione del sistema

Subito dopo l'esecuzione del kernel, viene eseguito lo script `init`, che legge il file `/etc/inittab` per scegliere la modalità di avvio. 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 filesystem, 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 la script `/etc/rc.d/rc.netdevice` per configurare la rete); `/etc/rc.d/rc.pcmcia` (che configura le periferiche di que-

sto 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 in modalità grafica (cioè all'interno del server X)

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 al n. **8.3.**) 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 creato 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); /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 installano, per poter funzionare, 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/cups (che gestisce le stampanti), /etc/rc.d/rc.gdm

(che abilita l'uso del mouse in console), /etc/rc.d/samba (che avvia SAMBA) 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 da lanciare 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**.

8.4. 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 del file.

8.5. Spegner Slackware

8.5.1. Di default, solo l'utente root può spegnere il computer. Se si è loggati come user, si deve terminare la sessione, passare a root col comando **su - root**, inserire la password di root e quindi lanciare il comando **shutdown -h now**.

8.5.2. Col kernel 2.4.x 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.x, se il supporto APM è abilitato nel kernel stesso, non serve effettuare alcun intervento: il computer si spegne da solo. Se si vuole che il computer si spenga dopo un determinato tempo, bisogna invece impartire il comando: **shutdown -h n**, dove *n* è il numero dei minuti dopo i quali avrà luogo lo spegnimento. Ad esempio, il comando **shutdown -h 30** ordina al sistema di spegnersi dopo 30 minuti; il comando **shutdown -h 0** è naturalmente lo stesso di **shutdown -h now**.

8.5.3. Perché anche gli utenti possano spegnere il computer è necessario usare l'utilità **sudo**. È sufficiente a questo fine aggiungere al file /etc/sudoers alle righe già presenti quelle sotto indicate, in modo che risulti così:

```
#Host alias specification
```

```
# User alias specification
```

```
User_Alias NOME_ALIAS = nome_utente, nome_utente, ...
```

```
# Cmnd alias specification
```

```
Cmnd_Alias NOME_COMANDI = /sbin/reboot, /sbin/shutdown
```

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

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 password per effettuare le operazioni. Tali utenti potranno spegnere il computer col comando **shutdown [-h n]** e riavviarlo col comando **reboot [-n]** con la seguente sintassi: **sudo /sbin/shutdown -h n** e **sudo /sbin/reboot [-n]**. È necessario inserire l'indirizzo completo, perché il comando risiede nella directory /sbin, inaccessibile agli utenti. Se poi volessimo abbreviare i comandi, potremmo sempre creare degli alias. In questo caso sarà necessario creare nella /home di ciascun utente il file .bash_profile (che di default Slackware non crea) includendovi le seguenti righe, che definiscono degli alias:

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

8.6. Italianizzare Slackware

Per italianizzare le applicazioni che non corrono sotto X, dobbiamo introdurre a mano le seguenti modifiche:

Nel file /etc/profile, all'inizio, alla voce # Set the values for some environment variables:

aggiungeremo le righe:

```
export LANG="it_IT@euro"
export CHARSET="iso8859-15".
```

Nel file /etc/profile.d/lang.sh

commenteremo la riga

```
export LANG=C
```

e aggiungeremo la riga

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

Infine nel file /etc/profile.d/lang.csh

commenteremo la riga

```
setenv LANG C
```

e aggiungeremo la riga:

```
setenv LANG it_IT.
```

8.7. Tradurre le man pages in italiano

Le man pages sono la prima, fondamentale risorsa per avere informazioni sui comandi di Linux. Digitando **man** seguito dal nome del comando su cui si vogliono ottenere le informazioni, comparirà la corrispondente pagina del manuale. Battendo invio si avanza di una riga alla volta, battendo il tasto spaziatore si avanza di un'intera schermata.

Una comunità italiana, il Progetto Pluto, ha realizzato la traduzione italiana della più gran parte delle man pages, per aiutare i non anglofoni. Essa è reperibile all'indirizzo: <http://ftp.pluto.linux.it/pub/pluto/ildp/man> e scaricare il file denominato man-pages-it-0.3.4.tar.gz o superiore. Si può quindi installare il file con la tradizionale procedura di compilazione, o compilandolo come *.tgz con **chec-**

kinstall. Le pagine tradotte saranno collocate nella directory `/usr/man/it`. Il loro utilizzo presuppone l'italianizzazione di Slackware che abbiamo descritto sopra, al n. **8.6**. Unico limite: la traduzione non è completa né sempre aggiornata rispetto alle man pages originali.

8.8. 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.
```

8.9. Installare font

8.9.1. Per installare in Slackware nuovi font TTF o Type1, ci sono svariate procedure. Il vecchio sottosistema **font/x** è ormai deprecato (abbandonato nelle ultime versioni di Slackware) e largamente sostituito dal nuovo sottosistema **font-config**, che consente prestazioni di visualizzazione decisamente migliori (come un avanzato anti-aliasing). Tuttavia il sottosistema Fontconfig non funziona ancora per alcune applicazioni come OpenOffice e Abiword, che usano una propria tecnologia di font rendering.

8.9.2. Col sottosistema **font/x** la procedura di aggiunta di font è la seguente:

- 1) copiare i font 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 font al server di font col comando **chkfontpath - -add /usr/share/fonts/local**.
- 3) Generare la lista dei font 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 font copiati, due nuovi file; `fonts.scale` e `fonts.dir`.
- 5) Riavviare il server dei font col comando **service xfs restart** o riavviare il server grafico.

8.9.3. Col recente sottosistema fontconfig la procedura di aggiunta globale di font è molto più semplice: è sufficiente copiare i fonts in una delle directory definite nelle opzioni di fontconfig, che è possibile trovare elencate nel file `/etc/fonts/fonts.conf`, in apertura, alla voce ``. Lanciare infine il comando **fc-cache /dir/dir/dir_dei_font** per aggiornare la cache delle informazioni dei font. Per aggiungere i font a un utente individuale, copiare i nuovi font nella directory `.fonts/` nella home directory dell'utente e quindi lanciare il comando **fc-cache -fv /dir_dei_font**. Il file di configurazione dei fonts è `/etc/fonts/fonts.conf`, che però non può essere modificato manualmente.

8.9.4. Poiché fontconfig è usato automaticamente per le applicazioni programmate usando il toolkit grafico Qt 3 o GTK+ 2, con KDE l'aggiunta di nuovi font può essere effettuata in modo molto semplice dal browser Konqueror trascinandolo nella directory `fonts:/` e relative subdirectory i file dei font da copiare. Il metodo più semplice resta comunque valersi dell'utilità Installatore dei tipi di carattere dal Centro di controllo di KDE.

8.9.5. Per migliorare l'anti-aliasing, serve una versione di freetype in cui l'anti-aliasing stesso sia abilitato. Il pacchetto è scaricabile all'indirizzo:
<http://www.slack.it/download/traverse.php?dir=.%2Flibrerie%2Ffreetype%2F2.1.9>.

8.10. Modificare i font della console

Per modificare i font 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 /etc/rc.d/rc.font inserendo dopo setfont -v il nome di uno dei font presenti in /usr/share/kbd/consolefonts.

8.11. Colorare la console

Perché nella console i vari tipi di file siano indicati con colori diversi, è necessario creare il file .bashrc nella directory base dell'utente (Slackware non lo crea di default) e inserirvi le seguenti righe:

```
export LS_OPTIONS='--color=auto'
eval `dircolors`
alias ls='ls $LS_OPTIONS'
alias ll='ls $LS_OPTIONS -l'
alias l='ls $LS_OPTIONS -lA'
```

In questo modo (che pure si può largamente personalizzare) si vedranno i nomi dei file in verde, quelli delle directory in blu, in rosso gli archivi compressi e i link in azzurro.

8.12. 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. Se, ad esempio, installeremo in seguito OpenOffice (o StarOffice), sarà necessario decomprimere il driver situato in /usr/share/ppd/marca_della_nostra_stampante e quindi installarlo seguendo la procedura di installazione da Ufficio - Gestione stampanti OpenOffice (o StarOffice). Infine, nella finestra delle proprietà della stampante, dovremo inserire il comando **lpr-cups**. Non ci resta adesso che stampare la pagina di prova.

8.13. Abilitare java in Mozilla

Per far sì che tutte le applicazioni trovino il plugin java, è necessario editare il file /etc/profile aggiungendo le seguenti righe (indichiamo qui la directory in cui Slackware installa di default Java):

```
JAVA_HOME=/usr/lib/java
PATH=$PATH:$JAVA_HOME/bin
MANPATH=$MANPATH:$JAVA_HOME/man
export JAVA_HOME PATH MANPATH.
```

In alternativa è possibile creare nella directory /etc/profile.d il file java.sh, contenente le stesse righe e renderlo eseguibile col comando **chmod a+x**. In tal modo non sarà necessario indicare ogni volta il percorso; ciò è utile ad esempio con OpenOffice e StarOffice, che si valgono anche di java per attivare determinati servizi. Le ultime versioni di Slackware fanno comunque per noi questo lavoro automaticamente.

Per abilitare java (qui ipotizziamo di avere installata la versione 1.4.2_4) in Mozilla, si deve creare manualmente un link simbolico nella directory dei plugin di Mozilla /usr/include/mozilla-1.6/plugin al plugin java, col comando

```
ln -s /usr/lib/j2sdk1.4.2_4/jre/plugin/i386/ns610-gcc32/libja-
```

`vaplugin_oji.so libjavaplugin_oji.so.`

8.14. 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 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
        true
    else
        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 StarOffice – Strumenti - Opzioni - Programmi ausiliari – Attiva i collegamenti con HTTP, indicando ovviamente l'intero percorso dello script.

8.15. 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 a ciò che intendete fare, potete rimuoverli modificando i permessi di `/etc/profile.d/bds-games-login-fortune.sh` e di `/etc/profile.d/bds-games-login-fortune.csh` col comando `chmod a-x`. In questo modo impedisce loro di essere eseguiti.

8.16. 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.conf`, `/etc/my.large.conf`, `/etc/my.medium.conf`, `/etc/my.small.conf` e copiarlo sempre in `/etc` rinominandolo `/etc/my.conf`. La scelta dipende dalle dimensioni che intendiamo attribuire al nostro database. Quindi bisogna lanciare in sequenza i seguenti comandi: 1) `mysql_install_db`, 2) `cd /var/lib`, 3) `chown -R mysql mysql/`, 4) `chgrp -R mysql mysql/`. In tal modo mySQL viene attivato. Eccede dalla presenta trattazione l'uso di questo database.

8.17. Installare il kernel 2.6.7

Slackware 10.0 adotta il kernel 2.4.26. Tuttavia nella directory `/testing/packages/linux-2.6.7` del CD 2 è presente anche il kernel 2.6.7. La sua installazione è estremamente semplice. Basta infatti effettuarla col comando `installpkg kernel-generic-2.6.7-i486-1.tgz`.

È opzionale installare anche i moduli e gli headers, sempre con la medesima procedura.

Tuttavia, prima di riavviare il sistema, bisogna compiere due operazioni. Innanzitutto creare `initrd`, in modo da caricare i moduli opportuni prima di avviare il kernel. La procedura è leggermente diversa a seconda del filesystem adottato

nella partizione di root. Se si tratta di ext3, si lancerà il comando **mkinitrd -c -k 2.6.7 -m jbd:ext3 -f ext3 -r /dev/hdan**

(ipotizzando che la partizione di root risieda in hda e dove *n* è il numero della partizione stessa). Se invece il filesystem è reiserfs, allora il comando risulta essere: **mkinitrd -c -k 2.6.7 -m reiserfs**.

Senza questo passaggio, i driver del filesystem non verrebbero caricati, producendo un blocco del sistema per "kernel panic".

In secondo luogo si deve aggiornare il file /etc/lilo.conf aggiungendo nella sezione # Linux bootable partition config begins le seguenti righe:

```
image = /boot/vmlinuz-generic-2.6.7
```

```
initrd = /boot/initrd.gz
```

```
root = /dev/hdan
```

```
label = Linux-2.6.7
```

```
read-only
```

Al solito, si deve lanciare **lilo** 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: di conseguenza, non sarà possibile ricompilare questo kernel senza installare separatamente, in un secondo momento, i sorgenti stessi.

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

8.18.1. Il kernel di Linux sopporta in maniera nativa l'accelerazione grafica. Tuttavia, risultati migliori si ottengono installando quelli specifici per la propria scheda video. Prenderemo qui in esame il caso delle schede ATI Radeon.

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>.

Vediamo la procedura per installare il driver originale ATI su Xorg (con Xfree-4.4 la sua installazione dava non pochi problemi, ora invece dà risultati certi). Innanzitutto nel kernel vanno disabilitate entrambe le voci che gestiscono l'accelerazione nativa, e cioè AGP e DRM. Inoltre devono essere disponibili nel sistema i sorgenti del kernel.

8.18.2. Il driver ATI è in formato *.rpm; trasformiamolo in *.tgz col comando **rpm2tgz fglrx-4.3.0-3.9.0.i386.rpm**, quindi installiamolo col comando **installpkg fglrx-4.3.0-3.9.0.i386.tgz**; il pacchetto dovrebbe risultare installato in /lib/modules/fglrx. Se usiamo il kernel 2.4.24 o abbiamo installato il 2.6 fino alla versione 6, possiamo passare alla fase successiva. Se invece abbiamo installato il kernel 2.6.7, ci aspetta un passaggio preliminare. A causa di una modifica nella dichiarazione di page, è necessario riportare alcune stringhe al valore riconosciuto dal driver (anteriore al kernel 2.6.7). Perciò dobbiamo aprire con un editor di testo prima il file /lib/modules/build_mod/agpgart_be.c e modificare le occorrenze di "count" (quando non preceduto da "." oppure da "_") in "_count"; quindi ripetere la medesima operazione per il file /lib/modules/build_mod/firegl_public.c. Si tratta di sette occorrenze nel primo file e di una sola nel secondo, in righe che iniziano sempre con "atomic_dec" o "atomic_inc".

8.18.3. A questo punto passiamo alla compilazione e installazione, che dev'essere effettuata senza che il server X sia attivo. Dovendo spostarci nella directory

```
/lib/modules/fglrx/build_mod, lanciamo in sequenza i seguenti comandi:  
cd /lib/modules/fglrx/build_mod (per spostarci nella directory dove com-  
pilare il driver)  
sh make.sh (e lavorerà l'ATI Module generator per compilare il driver)  
cd .. (per passare alla directory superiore)  
sh make_install.sh (per installare il driver).
```

8.18.4. Se tutto è andato a buon fine, possiamo provare a caricare il modulo col comando **modprobe fglrx**: col comando **lsmod** dovremmo vederlo caricato.

Per caricarlo all'avvio, dobbiamo lanciare l'utilità di configurazione **fglrxconfig**, oppure apportare alcune modifiche al preesistente file `/etc/X11/xorg.conf`. Nel primo caso, seguiremo un procedimento simile alla configurazione del server X già effettuato con **xorgconfig**, stando attenti a rispondere "no" quando il programma ci chiederà se vogliamo usare il driver nativo ("internal driver") del kernel. Il file di configurazione si chiama ancora `XF86Config-4`: lo rinomineremo pertanto `xorg.conf` col comando **mv XF86Config-4 xorg.conf**. Nel secondo caso dobbiamo apportare le seguenti modifiche in `/etc/X11/xorg.conf`:

alla Section Modules devono risultare decommentate le seguenti righe:

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

Alla sezione Graphic device section, Section "Device" è necessario sostituire il driver corrente (ad esempio "radeon") con "fglrx".

La sezione Section "DRI" (consistente di norma in tre righe) dev'essere decommentata per intero.

A questo punto rilanciamo il server X con **startx** e verificiamo 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 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** deve mostrarci i frames per secondo, con l'aiuto di una finestra grafica.

8.19. Aggiornare Slackware con Swaret

8.19.1. 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. Si tratta di **Swaret** (probabilmente il più noto), e quindi di **Slapt-get**. Qui prendiamo in considerazione il primo, che si sta rapidamente affermando come lo strumento più diffuso e fun-

zionale (tanto che ne esiste la traduzione in svariati linguaggi, italiano incluso, e anche una GUI per KDE chiamata `kswaret` che però appare un po' scomoda e non così necessaria, data la semplicità di gestire il programma da riga di comando. Slackware non installa Swaret di default, ma esso è reperibile nel secondo CD, fra gli extra, oppure scaricabile nella versione più aggiornata all'indirizzo <http://www.swaret.org>.

8.19.2. Dopo aver installato Swaret, è necessario configurarlo editando il file `/etc/swaret.conf.new` rinominandolo `/etc/swaret.conf`. Innanzitutto si devono inserire i repository che ci interessano, se vogliamo aggiungerne a quelli già presenti di default, con la seguente sintassi:

`ROOT=http://www.slack.it/download-$VERSION.`

Dobbiamo quindi controllare con un editor che almeno le seguenti righe, le più importanti fra quelle che definiscono le varie opzioni, risultino come descritto qui sotto:

`VERSION=10.0`: serve a stabilire la versione a cui aggiornare i pacchetti già presenti nel computer.

Dopo la riga `VERSION` è inoltre possibile aggiungere le seguenti righe:

`PACKAGES= 1` o `0`: effettua o meno la ricerca dei soli pacchetti standard.

`PATCHES= 1` o `0`: effettua o meno la ricerca delle patches.

`EXTRA= 1` o `0`: effettua o meno la ricerca dei pacchetti extra.

Inoltre:

`EXCLUDE`: esclude i pacchetti elencati dall'aggiornamento automatico; alcuni, come il kernel, sono presenti di default. Possiamo quindi aggiungerne altri.

`DEPENDENCIES=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=0`. Compie la ricerca delle librerie mancanti.

`DEP_INSTALL=1`. Effettua l'installazione effettuando il controllo delle dipendenze.

`DEP_UPGRADE=1`. Aggiorna le librerie mancanti.

Si può anche aggiungere la riga:

`DSEARCHARM = 1`

perché Swaret usi usare 'slocate' invece di 'find', il che velocizza la procedura.

In seguito, è necessario aggiornare il database delle librerie col comando **`updatedb`**. Poiché il database non viene creato di default, sarà preliminarmente necessario crearlo col comando **`updatedb -u`**.

8.19.3. A questo punto, per avviare l'aggiornamento vero e proprio, 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_del_pacchetto`**, oppure **`--remove`** o **`--upgrade`** rispettivamente per rimuoverlo o per aggiornarlo.

8.19.4. 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. In questo caso è necessario impostare la voce `VERSION=current`.

Un'ultima cautela: è meglio aggiornare alcuni pacchetti particolarmente delicati (come `Alsa`, `mySQL` o i compilatori) non servendosi di `Swaret`, che in certi casi dà errori, bensì manualmente, scaricandoli e aggiornandoli col comando `upgradedpkg`. Se si tratta di demoni già avviati, è necessario prima chiudere tutte le applicazioni in uso.

Maggiori dettagli sono reperibili nell'HOW-TO reperibile al seguente indirizzo: <http://www.sistemistiindipendenti.org/modules/news/article.php?storyid=68>.

8.20. Impostare un firewall

8.20.1. In Linux il firewall non è un'applicazione aggiuntiva, come in Windows, ma una funzionalità inserita nel kernel. Ciò lo rende particolarmente potente e versatile. Nelle versioni più recenti il firewall è `iptables` (che rimpiazza il vecchio `ipchain`). La sua configurazione manuale 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`. Questa utilità crea il file `/etc/rc.firewall`, che avvia automaticamente il firewall.

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**.

8.20.2. Nella linguetta **Zona** si possono configurare gli indirizzi IP locali (`Locale`), cioè quelli presenti se siamo in una rete, e gli indirizzi IP esterni (`Internet`), cioè quelli della navigazione: in altre parole, sia il traffico in uscita sia quello in entrata. Eventualmente, si possono anche inserire gli accessi (sotto forma di indirizzi IP oppure di DNS) ad alcuni protocolli e IP specifici creando una "nuova zona".

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 nella `Zona Internet` è spuntata la casella `Locale`, mentre nella `Zona locale` è spuntata la casella `Internet`.

Possiamo creare una nuova `Zona` nel caso vogliamo attivare un determinato protocollo, perché il nostro computer lo possa utilizzare per connettersi a `Internet`. 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, dovremmo 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.

8.20.3. 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.

8.20.3.1. Nella `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 abilitata 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 lasciatelo escluso) per poter gestire la posta elettronica.

Rete: 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 indispensabile 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 Peer-ToPeer.

Vari: è possibile abilitare qui il keyserver PGP per il controllo della posta elettronica e la cifratura dei messaggi.

8.20.3.2. 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.

8.20.4. 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.

8.20.5. Nell'ultima linguetta, **Avanzato**, oltre alla voce Disabilita il firewall (utile se dobbiamo riconfigurarla, magari a seguito di qualche errore grave di configurazione), 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.

Una volta terminata la configurazione, basta cliccare su Applica per salvare le modifiche apportate. Il firewall può a questo punto essere testato su alcuni siti appositi, come <http://www.pcflank.com>.