La guida Debian

Osamu Aoki <debian@aokiconsulting.com>
Traduzione italiana: Davide Di Lazzaro <mc0315@mclink.it>
'Autori' a pagina 137

CVS, Sun, 13 Oct 2002 22:40:17 -0600

Estratto

La guida Debian (http://qref.sourceforge.net/) copre molti aspetti dell'amministrazione di sistema attraverso esempi di comandi da shell. Vengono proposte lezioni base, trucchi e molte altre informazioni su molti argomenti, tra cui i concetti fondamentali tipici di un sistema Debian, suggerimenti per l' installazione, il sistema Debian di gestione dei pacchetti, il kernel sotto Debian, l'affinamento del sistema, la costruzione di un gateway, editor di testo, CVS, programmazione e GnuPG per i non-sviluppatori. Se cercate un aiuto per una manutenzione di emergenza del sistema andate immediatamente a 'Comandi di sopravvivenza Debian' a pagina 54. L'ultima versione ufficiale è http://www.debian.org/doc/manuals/debian-reference/ e l'ultima in fase di sviluppo è http://qref.sourceforge.net/Debian/. Il progetto è ospitato presso http://qref.sourceforge.net/. Esempi di script di configurazione possono essere trovati qui. (examples/)

Avviso di Copyright

Copyright © 2001-2002 by Osamu Aoki <debian@aokiconsulting.com> Copyright (Chapter 2) © 1996-2001 by Software in the Public Interest

Questa guida può essere utilizzata nei termini della GNU General Public License versione 2 o successive. (http://www.gnu.org/copyleft/gpl.html)

Permission is granted to make and distribute verbatim copies of this document provided the copyright notice and this permission notice are preserved on all copies.

Permission is granted to copy and distribute modified versions of this document under the conditions for verbatim copying, provided that the entire resulting derived work is distributed under the terms of a permission notice identical to this one.

Permission is granted to copy and distribute translations of this document into another language, under the above conditions for modified versions, except that this permission notice may be included in translations approved by the Free Software Foundation instead of in the original English.

Indice

1 Prefazione **1**

1.1 Convenzioni utilizzate 1

1.2 Impostazioni di base 2

1.3 Le basi delle distribuzioni Debian 2

2 Nozioni fondamentali della Debian **5**

2.1 Gli archivi Debian 5

2.1.1 Struttura della directory 5

2.1.2 Distribuzioni 6

2.1.3 La distribuzione `stable` 6

2.1.4 La distribuzione `testing` 7

2.1.5 La distribuzione `unstable` 7

2.1.6 La distribuzione `frozen` 8

2.1.7 Codice dei nomi della distribuzioni Debian 8

2.1.8 Nomi in codice usati in passato 8

2.1.9 Da dove vengono i nomi delle distribuzioni? 9

2.1.10 La directory `pool` 9

2.1.11 Alcune note storiche su `sid` 10

2.1.12 Pacchetti caricati in 'incoming' 10

2.1.13 Sezioni per architettura 10

2.1.14 Il codice sorgente 11

2.2 Il sistema di gestione dei pacchetti Debian 11
2.2.1 Panoramica dei pacchetti Debian 11
2.2.2 Il formato dei pacchetti Debian 12
2.2.3 Convenzioni nei nomi dei pacchetti Debian 13
2.2.4 Mantenimento della configurazione locale 13
2.2.5 Scripts di gestione Debian 14
2.2.6 Priorità 14
2.2.7 Pacchetti Virtuali 15
2.2.8 Dipendenze 15
2.2.9 Cosa significa Pre-Depends 16
2.2.10 Lo stato dei pacchetti 17
2.2.11 Evitare l'aggiornamento dei pacchetti 17
2.2.12 Pacchetti sorgente 18
2.2.13 Compilare pacchetti binari dai sorgenti 18
2.2.14 Creare nuovi pacchetti Debian 19
2.3 Aggiornare un sistema Debian 19
2.3.1 Metodi per aggiornare un sistema Debian 20
2.3.2 Panoramica degli strumenti di gestione dei pacchetti 20
2.3.3 dpkg 20
2.3.4 APT 21
2.3.5 dselect 21
2.3.6 Aggiornare un sistema in funzione 21
2.3.7 File .deb scaricati e tenuti in cache 21
2.3.8 Tenere una registrazione dell'aggiornamento 22
2.4 La sequenza di boot della Debian 22
2.4.1 `init` 22
2.4.2 Runlevels 22
2.4.3 Personalizzare il processo di boot 23
2.5 Supportare le differenze 24

2.6 Locale non in inglese . . . 24
2.7 Debian ed il kernel . . . 26
2.7.1 Compilare un kernel non debian . . . 26
2.7.2 Gli strumenti per compilare un kernel personalizzato. . . . 26
2.7.3 Boot loaders alternativi . . . 26
2.7.4 Boot floppy personalizzato . . . 27
2.7.5 Funzioni speciali per trattare con i moduli . . . 27
2.7.6 Disinstallare un vecchio pacchetto kernel . . . 27

3 Installazione di Debian **29**
3.1 Installazione di Linux . . . 29
3.1.1 Le basi della compatibilità hardware . . . 29
3.1.2 Definire i componenti del PC — il chipset . . . 30
3.1.3 Definire i componenti del PC — Debian . . . 30
3.1.4 Definire i componenti del PC — altri S.O. . . . 30
3.1.5 Il mito Lilo . . . 31
3.1.6 Scegliere i boot floppies . . . 31
3.1.7 Installazione . . . 31
3.1.8 Quale host ed IP usare per LAN . . . 32
3.1.9 Account utenti . . . 33
3.1.10 Partizionare il disco rigido ed impostare l'NFS . . . 33
3.1.11 Linee guida per la memoria DRAM . . . 35
3.1.12 Lo spazio di swap . . . 36
3.2 Configurare Bash . . . 36
3.3 Configurare il mouse . . . 37
3.4 Configurare NFS . . . 37
3.5 Configurare Samba . . . 38
3.6 Configurare la stampante . . . 38
3.7 Altri consigli di configurazione per l'host . . . 39

3.7.1 Installate pochi altri pacchetti ed attenetevi alla configurazione di base 39
3.7.2 Moduli 39
3.7.3 Impostazione base del CD-RW 39
3.7.4 Grosse memorie e spegnimento automatico 39
3.7.5 Altre modifiche di configurazione in /etc/* 40

4 Lezioni di Debian **41**
4.1 Fonti di informazione 41
4.2 La console in Linux 42
4.2.1 Login 42
4.2.2 Aggiungere un account utente 42
4.2.3 Lo spegnimento del PC 42
4.2.4 L'editing della riga di comando 43
4.2.5 Comandi base da ricordare 43
4.2.6 Il sistema X Window 44
4.2.7 Altre combinazioni di tasti 44
4.3 Midnight Commander (MC) 44
4.3.1 Installazione di MC 44
4.3.2 Lanciare MC 45
4.3.3 File manager 45
4.3.4 Trucchi per la riga di comando: 46
4.3.5 L'editor 46
4.3.6 Viewer 47
4.3.7 Auto start 47
4.3.8 File system virtuale FTP 47
4.4 Per studi ulteriori 48

5 Passare a Woody **49**
5.1 Preparazione 49

5.2 Aggiornamento a woody . . . 50
5.3 Configurare Woody . . . 51
5.4 Ottimizzare sources.list . . . 51

6 Gestione dei pacchetti in Debian **53**
6.1 Introduzione . . . 53
6.1.1 Strumenti fondamentali . . . 53
6.1.2 Strumenti utili . . . 54
6.2 Comandi di sopravvivenza Debian . . . 54
6.2.1 Installare i *task* con "`tasksel`" . . . 54
6.2.2 Installare il sistema con APT . . . 54
6.2.3 Aggiornare il sistema con APT . . . 55
6.2.4 Controllo dei bachi in Debian . . . 55
6.2.5 Aggiornamento con APT: risoluzione dei problemi . . . 56
6.2.6 Salvataggio mediante dpkg . . . 57
6.2.7 Installare un pacchetto su un sistema non bootabile . . . 57
6.3 Comandi del "paradiso" Debian . . . 58
6.3.1 Informazioni sui file . . . 58
6.3.2 Informazioni sui pacchetti . . . 58
6.3.3 Riconfigurare i pacchetti già installati . . . 59
6.3.4 Rimozione e purga dei pacchetti . . . 60
6.3.5 Mantenere vecchi pacchetti . . . 60
6.3.6 Dselect - configurazione generale . . . 61
6.3.7 Ridurre la dimensione dei files nella cache . . . 61
6.3.8 Salvare/copiare la configurazione del sistema . . . 61
6.3.9 Portare un pacchetto nel sistema "stable" . . . 62
6.3.10 Convertire od installare un pacchetto binario di altra distribuzione . . . 63
6.4 Altre particolarità di Debian . . . 63
6.4.1 dpkg-divert . . . 63

6.4.2 Pacchetto `equivs` . . . 63
6.4.3 Comandi alternativi . . . 64
6.4.4 Inizializzazione e livelli (runlevels) del System-V . . . 64
6.4.5 Demoni di servizio disabilitati . . . 65

7 Il kernel Linux su Debian **67**
7.1 Ricompilare il kernel . . . 67
7.1.1 Il metodo standard Debian . . . 67
7.1.2 Il metodo classico . . . 68
7.2 Kernel 2.4 modulare . . . 69
7.2.1 PCMCIA . . . 69
7.2.2 SCSI . . . 70
7.2.3 Funzioni di rete . . . 70
7.2.4 Il filesystem EXT3 (> 2.4.17) . . . 71
7.2.5 Supporto nel kernel 2.4 per Realtek RTL-8139 . . . 72

8 Trucchi per Debian **73**
8.1 Il sistema di boot . . . 73
8.1.1 Ho scordato la password di root (1) . . . 73
8.1.2 Ho scordato la password di root (2) . . . 74
8.1.3 Non riesco a fare il boot . . . 74
8.1.4 Altri trucchi per il boot prompt . . . 74
8.2 Registrazione degli eventi . . . 75
8.2.1 Registrare gli eventi della shell . . . 75
8.2.2 Registrare gli eventi di X . . . 76
8.3 Copia ed archiviazione di una intera subdirectory . . . 76
8.3.1 Comandi base per copiare una intera subdirectory . . . 76
8.3.2 cp . . . 76
8.3.3 tar . . . 77

8.3.4 pax . . . 77
8.3.5 cpio . . . 77
8.3.6 afio . . . 77
8.4 Modificare i files con le espressioni regolari . . . 78
8.5 Recuperare un sistema bloccato . . . 78
8.5.1 Uccidere un processo . . . 78
8.5.2 ALT-SysRQ . . . 78
8.6 Files di configurazione . . . 79
8.6.1 Controllo del login . . . 79
8.6.2 Defaults . . . 79
8.7 Alcuni piccoli comandi da ricordare . . . 79
8.7.1 Memoria libera . . . 79
8.7.2 Impostare l'ora (BIOS) . . . 80
8.7.3 Come disabilitare lo screensaver . . . 80
8.7.4 Disabilitare il suono (beep) . . . 81
8.7.5 Messaggi di errore sulla console . . . 81
8.7.6 Riportare la console allo stato normale . . . 81
8.7.7 Convertire file di testo da DOS a Unix . . . 81
8.7.8 Convertire grossi files in files più piccoli . . . 82
8.7.9 Parti di script per il piping . . . 82
8.7.10 Cattura il testo / archivio delle ML da una pagina web. . . . 82
8.7.11 Il comando time . . . 82
8.7.12 Il comando nice . . . 83
8.7.13 Programmare gli eventi (cron, at) . . . 83
8.7.14 CScambio di console con `screen` . . . 84
8.7.15 Testare la rete: le basi . . . 85
8.7.16 Eliminare la posta dallo spool locale . . . 85
8.7.17 Rimuovere la posta "congelata" (bloccata) dallo spool locale . . . 85
8.7.18 Pulire il contenuto di un file . . . 85

8.7.19 Files fantoccio . . . 86
8.7.20 chroot . . . 86
8.7.21 Samba . . . 86

9 Perfezionare Debian **87**
9.1 sudo - un ambiente di lavoro più sicuro . . . 87
9.2 Backup . . . 88
9.3 Masterizzatori-writer . . . 88
9.3.1 Introduzione . . . 88
9.3.2 Approccio 1: (modules+lilo) . . . 88
9.3.3 Approccio 2: (ricompilare il kernel): . . . 89
9.3.4 Fasi post-configurazione . . . 89
9.3.5 Il file CD-image (bootabile) . . . 90
9.3.6 Scrivere sul masterizzatore (R, R/W) . . . 90
9.3.7 Fare un'immagine di un CD . . . 91
9.3.8 Creare un CD Debian . . . 91
9.3.9 Backup del sistema su CD-R . . . 92
9.3.10 Copiare CD musicali su CD-R . . . 92
9.4 I programmi X . . . 92
9.4.1 Il server X . . . 92
9.4.2 Client X . . . 93
9.4.3 Connessione TCP/IP ad X . . . 93
9.4.4 Connessione X remota: Xhost . . . 94
9.4.5 Connessione X remota: SSH . . . 94
9.4.6 Xterm . . . 94
9.4.7 Diventare root in X . . . 94
9.5 SSH . . . 95
9.5.1 Principi basilari . . . 95
9.5.2 Port forwarding - per il tunneling smtp/pop3 . . . 95

9.5.3 Connettersi con meno passwords possibili - RSAAuthentication 96
9.5.4 Problemi 96
9.6 Programmi di gestione posta 97
9.6.1 Mail transport agent (Exim) 97
9.6.2 Raccolta di tutti gli indirizzi e-mail inesistenti (Exim) 97
9.6.3 Utilità per la posta (fetchmail) 98
9.6.4 Utilità per la posta (procmail) 98
9.6.5 Mail user agent (Mutt) 98

10 Costruire un gateway con Debian **101**
10.1 Configurazione di rete 101
10.1.1 Configurazione dell'host 101
10.1.2 IP-masquerade 102
10.1.3 Punti principali di una configurazione di rete 103
10.2 Gestione di connessioni multiple di rete 103

11 Editor **105**
11.1 Editor di testo 105
11.2 Emacs e Vim 105
11.2.1 Comandi utili per Vim 105
11.2.2 Comandi utili per Emacs 106
11.2.3 Lanciare l'editor 106
11.2.4 Riassunto dei comandi per emacs e vim 106
11.2.5 Configurare Vim 109
11.2.6 Ctags 109
11.2.7 Convertire una schermata da "sintassi-evidenziata" in sorgente HTML 109
11.2.8 Dividere lo schermo con `vim` 110

12 CVS **111**

12.1 Installare un server CVS . . . 111

12.2 Esempi di una sessione CVS . . . 112

12.2.1 CVS anonimo (solo download) . . . 112

12.2.2 Uso del server CVS locale . . . 112

12.2.3 Uso di un pserver CVS remoto . . . 112

12.2.4 Uso di un CVS remoto tramite ssh . . . 112

12.2.5 Creare un nuovo archivio CVS . . . 112

12.2.6 Lavorare con CVS . . . 113

12.2.7 Esportare i files da CVS . . . 114

12.2.8 Amministrare CVS . . . 114

12.3 Risoluzione dei problemi . . . 114

12.3.1 Permessi dei file nel deposito . . . 114

12.3.2 Il bit d'esecuzione . . . 115

12.4 Comandi CVS . . . 115

13 Programmazione **117**

13.1 Dove iniziare . . . 117

13.2 BASH . . . 117

13.3 AWK . . . 118

13.4 PERL . . . 119

13.5 PYTHON . . . 120

13.6 MAKE . . . 121

13.7 C . . . 122

13.7.1 Un semplice programma in C (gcc) . . . 123

13.7.2 Debugging (gdb) . . . 124

13.7.3 Flex – una miglior. . . Lex . . . 124

13.7.4 Bison – un Yacc migliore . . . 125

13.7.5 Autoconf – disinstallare . . . 125

13.8 SGML . . . 125
13.9 Creare pacchetti debian . . . 126

14 GnuPG **127**
14.1 Installazione di GnuPG . . . 127
14.2 Uso di GnuPG . . . 128
14.3 Gestione di GnuPG . . . 128
14.4 Uso con Mutt . . . 129

15 Supporto per Debian **131**
15.1 Riferimenti . . . 131
15.2 Trovare il significato di una parola . . . 134
15.3 Il sistema Debian di tracciamento dei bachi . . . 134
15.4 Mailing lists . . . 134
15.5 Internet Relay Chat . . . 134
15.6 Motori di ricerca . . . 135
15.7 Siti Web . . . 135

A Appendice **137**
A.1 Autori . . . 137
A.2 Garanzie . . . 139
A.3 Feedback . . . 139
A.4 Reperibilità . . . 139
A.5 Formato del documento . . . 140
A.6 Il labirinto di Debian . . . 140

Capitolo 1

Prefazione

Questo documento ha avuto origine come un manuale di riferimento “rapido”, per cui le informazioni che vi sono contenute possono essere sotto forma di veloci richiami o di indirizzi a riferimenti più autorevoli elencati in ‘Riferimenti’ a pagina 131.

1.1 Convenzioni utilizzate

“La guida Debian” fornisce informazioni attraverso brevi comandi BASH. Le convenzioni utilizzate sono le seguenti:

```
# comando nel root account
$ comando nello user account ... descrizione dell'azione
```

Per quanto attiene a BASH, vedere inoltre `man bash` e Bash-Prog-Intro-HOWTO (LDP) per iniziare

Abbreviazioni:

- LDP: Linux Documentation Project `http://www.tldp.org/`
- DDP: Debian Documentation Project `http://www.debian.org/doc/`

Gli esempi degli scripts si trovano qui. I file nascosti preceduti da . sono stati convertiti in file preceduti da _ (`examples/`).

1.2 Impostazioni di base

Se il sistema è stato installato con l'opzione "Simple", selezionate l'opzione "New user documentation". Altrimenti, date i seguenti comandi:

```
# dselect update tasksel ... seleziona l'opzione "New user documentation" ed
  altre simili
```

In aggiunta, installate questi pochi altri pacchetti tramite:

```
# apt-get install debian-policy developers-reference maint-guide \
       packaging-manual doc-debian doc-linux-text info \
       man-db manpages manpages-dev less mc # for potato

# apt-get install debian-policy developers-reference maint-guide \
       doc-debian doc-linux-text apt-howto info \
       man-db manpages manpages-dev less mc # for woody
```

1.3 Le basi delle distribuzioni Debian

Debian è disponibile in tre versioni:

- stable: Ottima per organizzare un server. Noioso se volete mettere su una workstation (WS). 'La distribuzione `stable`' a pagina 6
- testing: Valida per una WS. 'La distribuzione `testing`' a pagina 7
- unstable: Non installatela nè aggiornatela alla cieca. 'La distribuzione `unstable`' a pagina 7

Leggete almeno la mailing list chiave `debian-devel-announce@lists.debian.org` per gli aggiornamenti sullo stato della Debian. Allo stato attuale (marzo 2002), queste versioni corrispondono a Potato (stabilità a livello di produzione), Woody (in beta test, ora molto stabile) e sid (alfa test). Quando i pacchetti in unstable non ricevono nessun annuncio di bug grave entro una settimana o giù di lì, vengono automaticamente promossi a testing. Vedere 'Gli archivi Debian' a pagina 5.

In teoria, potete seguire due strade per ottenere il software più aggiornato.

'Installare il sistema con APT' a pagina 54: Per il lato WS.

'Portare un pacchetto nel sistema "stable"' a pagina 62: Per il lato server.

Dopo aver spiegato alcuni concetti fondamentali della distribuzione Debian 'Nozioni fondamentali della Debian' a pagina 5 presenterò alcune informazioni base per vivere felicemente con il software più aggiornato, unendo i vantaggi delle distribuzioni testing ed unstable. Per i più impazienti, andate direttamente al paragrafo 'Comandi di sopravvivenza Debian' a pagina 54. Felice aggiornamento!

Capitolo 2

Nozioni fondamentali della Debian

Questo capitolo spiega un sistema Debian a partire dai suoi fondamentali, ed è indirizzato ai non-sviluppatori. Per avere informazioni più autorevoli, vedere:

- Debian Policy Manual
- Debian Packaging Manual (potato)
- Debian Developer's Reference
- Debian New Maintainers' Guide

reperibili sotto 'Riferimenti' a pagina 131.

Se state cercando una qualsiasi risposta che li riguarda senza, però, tutti i loro dettagli,andate direttamente a 'Gestione dei pacchetti in Debian' a pagina 53 o ad altri capitoli.

Questo capitolo è formato da documenti presi dalla "Debian FAQ", e profondamente riorganizzati, per permettere ad un qualsiasi amministratore di un sistema Debian di avere un solido punto di partenza.

2.1 Gli archivi Debian

2.1.1 Struttura della directory

Il software impacchettato per la debian, è disponibile in uno dei numerosi alberi directory su ciascun Debian mirror site (`http://www.debian.org/misc/README.mirrors`) raggiungibili tramite FTP o HTTP.

Queste sono le directories presenti su ciascun mirror, sotto la directory `/debian/`:

/*dists*/: Contiene le "distribuzioni" ed era il luogo canonico di accesso dei pacchetti disponibili nelle versioni rilasciate e pre-rilascio. Alcuni vecchi pacchetti ed i files `Packages.gz` sono ancora qui.

/*pool*/: Nuova locazione, che contiene fisicamente tutti i pacchetti, sia quelli della versione rilasciata, che quelli pre-rilascio.

/*tools*/: Utilità DOS per creare dischetti boot, partizionare il disco rigido, comprimere/decomprimere i files e lanciare Linux.

/*doc*/: La documentazione base, come le FAQ, le istruzioni per la notifica dei bachi, ecc.

/*indices*/: I files dei Manutentori, ed i files override.

/*project*/: In gran parte materiale solo per sviluppatori, tipo:

***project/experimental/*:** Pacchetti e strumenti ancora in via di sviluppo, in fase alfa. I normali utenti non dovrebbero utilizzare i pacchetti qui contenuti, che possono essere pericolosi persino per i più esperti.

***project/orphaned/*:** Pacchetti lasciati dai loro vecchi manutentori e tolti dalla distribuzione.

2.1.2 Distribuzioni

Di norma sono tre le distribuzioni contenute nella directory `dists`. Sono definite come la distribuzione "stable", la "testing" e la "unstable". Talvolta se ne aggiunge una quarta, la "frozen" (vedere 'La distribuzione `frozen`' a pagina 8). Ogni distribuzione viene definita con un link simbolico alla directory reale, tramite un nome proprio nella directory `dists`.

2.1.3 La distribuzione `stable`

E' contenuta nella directory `stable`:

- stable/main/: Contiene i pacchetti che costituiscono formalmente il rilascio più recente del sistema.

  Tutti i pacchetti sono totalmente complianti con le Debian Free Software Guidelines (`http://www.debian.org/social_contract#guidelines`) e sono utilizzabili e distribuibili liberamente.

- stable/non-free/: Contiene i pacchetti la cui distribuzione è in qualche modo limitata, tale da richiedere ai distributori delle cautele dovute ai loro requisiti specifici di copyright.

  Per esempio alcuni pacchetti hanno licenze che ne vietano la distribuzione commerciale. Altri possono essere ridistribuiti, ma sono degli shareware di fatto, e non freeware. Prima che

tali pacchetti possano essere inclusi in qualsiasi ridistribuzione (come un CD-ROM, p.es.), le loro licenze devono essere studiate e possibilmente rinegoziate.

- stable/contrib/: Contiene i pacchetti che sono di per sè DFSG-free e *liberamente distribuibili*, ma dipendono in qualche modo da un pacchetto che *non è* liberamente distribuibile, ed è quindi disponibile nella sezione non-free.

Lo stato attuale della distribuzione `stable` è riportato in: stable problems (http://ftp-master.debian.org/testing/stable_probs.html).

2.1.4 La distribuzione `testing`

I pacchetti vengono inseriti nella directory 'testing' dopo aver subito un periodo di test nella unstable. I nuovi pacchetti sono contenuti nella directory pool 'La directory `pool`' a pagina 9. La directory testing contiene anche le sottodirectory main, contrib e non-free, divise secondo gli stessi criteri della 'stable'.

I pacchetti devono essere sincronizzati in tutte le architetture per le quali sono stati compilati e non devono mostrare dipendenze tali da renderli non installabili; devono inoltre avere meno bachi release-critical delle versioni sotto test. In questo modo si auspica che 'testing' sia sempre molto vicina ad essere candidata al rilascio. Per maggiori dettagli sul meccanismo che regola la distribuzione vedere http://ftp-master.debian.org/testing/.

Lo stato aggiornato della distribuzione `testing` è riportato presso:

- update excuses (http://ftp-master.debian.org/testing/update_excuses.html)
- testing problems (http://ftp-master.debian.org/testing/testing_probs.html)
- Release-critical bugs (http://bugs.debian.org/release-critical/)
- Base system bugs (http://base.debian.net/)
- Bugs in standard and task packages (http://standard.debian.net/)
- Other bugs and bug squashing party notes (http://bugs.debian.net/)

2.1.5 La distribuzione `unstable`

I pacchetti vengono inseriti nella directory 'unstable' dopo essere stati caricati nell'archivio Debian e quivi rimangono finchè non vengono trasferiti in 'testing' dopo qualche tempo. I nuovi pacchetti sono contenuti nella directory pool 'La directory `pool`' a pagina 9. La directory 'unstable' contiene anche le sottodirectory main, contrib e non-free, divise secondo gli stessi criteri della 'stable'.

La distribuzione 'unstable' contiene le immagini più recenti del sistema in fase di sviluppo. Gli utenti possono liberamente usare e testare questi pacchetti, ma vengono avvisati del loro precario stato di preparazione. Il vantaggio di usare la distribuzione unstable è quello di essere sempre al massimo dell'aggiornamento Debian, siate però pronti a raccogliere i pezzi se qualcosa va storto.

Lo stato aggiornato della distribuzione `unstable` è riportato presso: unstable problems (http://ftp-master.debian.org/testing/unstable_probs.html).

2.1.6 La distribuzione `frozen`

Una volta che la distribuzione testing è sufficientemente matura, diventa frozen; ciò significa che nessun nuovo codice viene più accettato, solo eliminazioni di bachi, se necessari. In aggiunta un nuovo albero testing viene creato nella directory `dists`, con un nuovo nome. La distribuzione frozen passa attraverso un ciclo di test (chiamato appunto 'test cycles') di qualche mese caratterizzato da aggiornamenti intermittenti ed importanti stabilizzazioni.

Viene tenuto un registro dei bachi della distribuzione frozen che possono impedire il rilascio di un pacchetto o di tutta la distribuzione. Una volta che il conteggio dei bachi scende al di sotto di una valore massimo prestabilito, la distribuzione frozen diventa stable e viene rilasciata. La precedente distribuzione stable diventa obsoleta e finisce in archivio.

2.1.7 Codice dei nomi della distribuzioni Debian

I nomi delle directory localizzate fisicamente nella directory `dists`, come `potato` e `woody`, sono semplicemente dei nomi in codice. Quando una distribuzione è nella fase di sviluppo le viene assegnato un nome in codice e non un numero di versione. Lo scopo di questi nomi è di rendere il mirroring delle distribuzioni più semplice (se, ad esempio, una directory reale come `unstable` cambiasse improvvisamente di nome in `stable`, una gran quantità di programmi dovrebbe essere nuovamente scaricata senza motivo).

Attualmente `stable` è un link simbolico a `potato` (i.e. Debian 3.0r0) e `testing` è un link simbolico a `woody`. Ciò significa che `potato` è la distribuzione attualmente stable e `woody` è l'attuale testing.

`unstable` è un link simbolico permanente a `sid`, dato che `sid` è sempre la distribuzione unstable.

2.1.8 Nomi in codice usati in passato

I nomi in codice che sono già stati utilizzati sono: `buzz` per la release 1.1, `rex` per la 1.2, `bo` per la 1.3.x, `hamm` per la 2.0 e `slink` per la 2.1.

Mentre le distribuzioni più recenti sono contenute nella directory `debian` su ciascun mirror, gli archivi delle vecchie distribuzioni, tipo`slink` sono contenuti nella directory `debian-archive`, sempre su ciascun mirror.

2.1.9 Da dove vengono i nomi delle distribuzioni?

Finora sono stati presi dai nomi dei personaggi del film“ Toy Story” della Pixar.

- *buzz* (Buzz Lightyear) era l'astronauta,
- *rex* era il tirannosauro,
- *bo* (Bo Peep) era la bambina che si prese cura della pecorella,
- *hamm* era il porcellino salvadanaio,
- *slink* (Slinky Dog) era il cane giocattolo,
- *potato* era, ovviamente, Mr. Potato,
- *woody* era il cowboy.
- *sid* era il bambino della porta accanto che rompeva i giocattoli.

2.1.10 La directory `pool`

Storicamente i pacchetti erano contenuti nella subdirectory di `dists` corrispondente alla distribuzione di cui facevano parte. Questo portò a vari problemi, tipo un grosso consumo di banda di connessione dei mirror ogni volta che venivano fatti dei cambiamenti di grossa entità.

Ora i pacchetti vengono tenuti in una grossa “vasca” (pool), strutturata in accordo con il nome del pacchetto sorgente. Per rendere il tutto maneggevole, la vasca è suddivisa in sezioni (main, contrib e non-free) e per la prima lettera del nome del pacchetto sorgente. Queste directory contengono svariati files: binari per ciascuna architettura ed i pacchetti sorgente da cui i pacchetti binari sono stati generati.

E' possibile sapere dove ciascun pacchetto è situato eseguendo un comando tipo: `apt-cache showsrc nomemiopacchetto` ed andando a leggere la riga 'Directory:'. Per esempio, i pacchetti `apache` sono immagazzinati in `pool/main/a/apache/`. Essendo molteplici, i pacchetti `lib*` vengono trattati in maniera particolare: per esempio, i pacchetti libpaper sono immagazzinati in `pool/main/libp/libpaper/`.

Le directory `dists` vengono ancora utilizzate per i file indice usati da programmi tipo `apt`. Inoltre, al momento attuale le vecchie distribuzioni non sono state convertite ad usare le vasche,

per cui si troveranno i percorsi contenenti distribuzioni tipo potato o woody nel campo Filename dell'intestazione.

Di norma non avete da preoccuparvi di ciò, poichè il nuovo `apt` e probabilmente il vecchio `dpkg-ftp` (vedere 'Metodi per aggiornare un sistema Debian' a pagina 20) sono in grado di gestire la cosa senza problemi. Se volete maggiori informazioni, andate a vedere Debian Package Pools FAQ (`http://people.debian.org/~joeyh/poolfaq`).

2.1.11 Alcune note storiche su `sid`

Quando il sid attuale non esisteva, l'organizzazione dell'archivio Debian aveva un problema principale: l'assunto che quando un'architettura veniva creata nell'unstable attuale, sarebbe stata rilasciata quando la distribuzione diventava la nuova stable. Però per molte architetture questo non è il caso, con il risultato che quelle directory dovevano essere mosse al momento del rilascio. Fatto poco pratico, poichè lo spostamento avrebbe fagocitato grosse quantità di banda.

Gli amministratori dell'archivio hanno evitato questo problema per pacchetti anni piazzando i binari delle architetture ancora non rilasciate in una directory speciale chiamata "sid". Al rilascio esisteva un link dall'architettura a quel momento stable a sid e da quel momento in poi essa veniva creata all'interno dell'albero unstable, come di norma. Tutto ciò era motivo di confusione per gli utenti.

Con l'avvento della vasca dei pacchetti (vedere 'La directory `pool`' nella pagina precedente) durante lo sviluppo della distribuzione "woody" i pacchetti binari cominciarono ad essere immagazzinati in una locazione canonica nella vasca, indipendentemente dalla distribuzione; in tal modo il rilascio di una distribuzione non determina più la grossa dispersione di banda sui mirror (c'è, ovviamente, un notevole consumo, ma graduale, di banda durante la fase di sviluppo).

2.1.12 Pacchetti caricati in 'incoming'

I pacchetti che vengono caricati nell'archivio vengono dapprima immagazzinati in `http://incoming.debian.org/` prima di accertarsi che provengano realmente da uno sviluppatore Debian. una volta controllati, vengono mossi da `incoming` ad `unstable`.

In caso di emergenza, potreste voler installare i pacchetti da qui, prima che raggiungano `unstable`. Li potete scaricare manualmente e controllare la firma GPG e le MD5sums nei files .changes e .dsc files, poi installarli.

2.1.13 Sezioni per architettura

All'interno di ciascun albero directory principale (`dists/stable/main`, `dists/stable/contrib`, `dists/stable/non-free` e `dists/unstable/main/`, etc.), i nomi dei pacchetti binari risiedono all'interno di subdirectories i cui nomi indicano l'architettura per la quale sono stati compilati.

- binary-all/, per pacchetti architettura-indipendenti. Comprendono, per esempio, scripts Perl o pura documentazione.
- binary-.../, per pacchetti che girano su una particolare piattaforma.
- ...

Ricordate che i reali pacchetti binari per *testing* ed *unstable* non risiedono più in queste directory, ma al livello principale della directory `pool` . I file elenco (Packages e Packages.gz) sono stati comunque mantenuti, per compatibilità con il vecchio sistema.

Per sapere quali architetture sono al momento supportate, leggetevi le Note di Rilascio per ciascuna distribuzione. possono essere trovate presso stable Release Notes links (`http://www.debian.org/releases/stable/releasenotes`) e testing Release Notes links (`http://www.debian.org/releases/testing/releasenotes`).

2.1.14 Il codice sorgente

Il codice sorgente è disponibile per ogni cosa contenuta nel sistema Debian. In più, i termini di licenza della maggior parte dei programmi *richiedono* che il codice venga distribuito insieme ai programmi, o che un'offerta di fornitura del codice li accompagni.

Di regola il codice viene reperito nelle directory "source", che sono in parallelo a tutte le directory dei binari architettura-specifiche, o più di recente alla directory `pool` vedere 'La directory `pool`' a pagina 9). Per scaricare il codice sorgente senza la necessità di essere addentro alla struttura dell'archivio Debian, provate un comando tipo `apt-get source nomemiopacchetto`.

Alcuni pacchetti, in particolare `pine`, sono disponibili solamente come sorgenti, a causa delle limitazioni delle licenze. Recentemente è stato fornito per la bisogna il pacchetto `pine-tracker` . Le procedure descritte in 'Portare un pacchetto nel sistema "stable"' a pagina 62 e 'Creare pacchetti debian' a pagina 126 dovrebbero fornire tutto il necessario per compilare un pacchetto manualmente.

Il codice sorgente potrebbe non essere disponibile, invece, per i pacchetti delle directory "contrib" e "non-free", che formalmente non fanno parte del sistema Debian.

2.2 Il sistema di gestione dei pacchetti Debian

2.2.1 Panoramica dei pacchetti Debian

Normalmente i pacchetti contengono tutti i files necessari all'implementazione di una serie di comandi o di funzionalità. Esistono due tipi di pacchetti:

- *Pacchetti binari*, che contengono eseguibili, file di configurazione, pagine man/info, informazioni sul copyright ed altra documentazione. Questi pacchetti vengono distribuiti in un formato specifico alla Debian (vedere 'Il formato dei pacchetti Debian' in questa pagina); si riconoscono per il suffisso .deb. Questi pacchetti possono essere "spacchettati" usando l'utilità tutta Debian `dpkg`; i dettagli si possono vedere alla pagina man corrispondente.
- *Pacchetti sorgente*, che consistono in un file `.dsc` che descrive il pacchetto sorgente (inclusi in nomi dei file seguenti), un file `.orig.tar.gz` che contiene i sorgenti originali non modificati in formato tar gzip ed in genere un file `.diff.gz` che contiene le modifiche specifiche per Debian ai sorgenti originali. L'utilità `dpkg-source` impacchetta e spacchetta questo tipo di pacchetti. Per i dettagli, ovviamente, la pagina man corrispondente.

L'installazione del software attraverso il sistema dei pacchetti utilizza delle "dipendenze", che sono state accuratamente costruite dal responsabile (manutentore) del pacchetto. Le dipendenze vengono descritte nel file `control`, associato a ciascun pacchetto. Ad esempio, il pacchetto contenente il compilatore GNU C (`gcc`) "dipende" dal pacchetto `binutils` che include il linker e l'assembler. Se si prova ad installare `gcc` senza aver prima installato `binutils`, il sistema di gestione dei pacchetti (dpkg) invierà un messaggio di errore riguardo alla necessità di avere anche `binutils` e bloccherà l'installazione di `gcc`. (Questo comportamento può comunque essere scavalcato dall'utente tenace, vedere al riguardo `dpkg(8)`.) Vedere più sotto in 'Dipendenze' a pagina 15.

Gli strumenti Debian per la gestione dei pacchetti possono essere usati per:

- manipolare e gestire i pacchetti o parte di essi,
- aiutare l'utente nella frammentazione dei pacchetti che devono essere trasmessi con un mezzo di limitate capacità come un floppy,
- aiutare gli sviluppatori nella costruzione degli archivi dei pacchetti e
- aiutare gli utenti nell'installazione dei pacchetti residenti in un archivio remoto Debian.

2.2.2 Il formato dei pacchetti Debian

Un "pacchetto" Debian, od un file dell'archivio Debian contiene gli eseguibili,le librerie e tutta la documentazione associata ad un gruppo o suite di programmi correlati. I file dell'archivio Debian, di norma, hanno il suffisso `.deb`.

I dettagli dei pacchetti binari Debian sono descritti nella pagina man `deb(5)`. Il loro formato interno è soggetto a cambiamenti (tra una versione maggiore e l'altra di Debian), per cui leggete sempre `dpkg-deb(8)` prima di manipolare i `.deb` files.

Almeno fino a "woody", gli archivi Debian sono sempre stati manipolabili anche dai normali comandi Unix, tipo `ar` e `tar`, anche quando i comandi dpkg non erano disponibili.

2.2.3 Convenzioni nei nomi dei pacchetti Debian

La nomina di un pacchetto Debian segue la regola seguente:

```
<foo>_<NumeroVersione>-<NumeroRevisioneDebian>.deb
```

`foo` sta per il nome del pacchetto. Come prova, si può risalire al nome del pacchetto associato ad un archivio Debian particolare (file .deb) in uno dei seguenti modi:

- ispezionare il file "Packages" nella directory dove è stato archiviato in un qualsiasi archivio Debian. Questo file contiene una sezione che descrive ciascun pacchetto presente; il primo campo di ciascuna sezione rappresenta il nome formale del pacchetto.
- utilizzando il comando `dpkg --info foo_VVV-RRR.deb` (dove VVV e RRR sono rispettivamente la versione e la revisione del pacchetto in questione). Il comando mostra, fra le altre cose, il nome del pacchetto corrispondente al file dopo lo spacchettamento.

`VVV` rappresenta il numero di versione specificato dallo sviluppatore principale. Non esiste uno standard, per cui il numero può presentarsi in formati diversi, tipo "19990513" e "1.3.8pre1".

`RRR` rappresenta il numero di revisione Debian e viene specificato dallo sviluppatore Debian (o da un utente qualsiasi, se decide di costruirsi il pacchetto da sè). Il numero corrisponde al livello di revisione del pacchetto Debian, quindi un nuovo numero in genere significa dei cambiamenti nel Makefile Debian (`debian/rules`), nel file di controllo Debian (`debian/control`), negli scripts di installazione o rimozione (`debian/p*`) oppure nei file di configurazione utilizzati con il pacchetto.

2.2.4 Mantenimento della configurazione locale

Il mantenimento di files configurabili dall'utente viene ottenuto tramite il meccanismo dei "conffiles" Debian. I file di configurazione dell'utente (di norma inseriti in `/etc`) vengono specificati nei `conffiles` all'interno del sistema dei pacchetti Debian. Il sistema di gestione dei pacchetti garantisce che, all'aggiornamento, i file di configurazione non vengano sovrascritti.

Per determinare esattamente quali files saranno preservati durante un aggiornamento, lanciare:

```
dpkg --status package
```

E guardare sotto "Conffiles:".

Le specifiche riguardo al contenuto dei conffiles Debian si trovano nel Debian Policy Manual, sezione 11.7, vedere 'Riferimenti' a pagina 131.

2.2.5 Scripts di gestione Debian

Gli scripts di gestione Debian sono degli script eseguibili che vengono lanciati automaticamente prima o dopo l'installazione di un pacchetto. Insieme ad un file chiamato `control`, tutti questi files fanno parte della sezione "control" di un file Debian.

I singoli files sono:

preinst Questo script viene eseguito prima che il pacchetto venga estratto dal file Debian (".deb"). Molti script 'preinst' interrompono i servizi per i pacchetti che devono essere aggiornati fino a che la loro installazione o aggiornamento non sono completati (a seguire dell'esecuzione con successo dello script 'postinst').

postinst Questo script tipicamente completa ogni configurazione richiesta dal pacchetto `foo` dopo che `foo` è stato estratto dal suo file Debian (":deb"). Spesso gli scripts 'postinst' richiedono all'utente determinate azioni e/o lo avvertono che, qualora accettasse le impostazioni di base, deve ricordarsi di riconfigurare il pacchetto se la situazione lo richiede. Molti scripts 'postinst', poi, eseguono tutti i comandi necessari a lanciare o far ripartire i servizi, dopo che il pacchetto è stato aggiornato o installato.

prerm Questo script ferma tutti i demoni associato con un pacchetto. Viene eseguito prima della rimozione di files associati ad un determinato pacchetto.

postrm Modifica i links od altri files correlati a `foo`, e/o rimuove i files creati da quel pacchetto.(Vedere anche 'Pacchetti Virtuali' a fronte.)

Tutti i file di controllo possono essere localizzati nella directory `/var/lib/dpkg/info`. I files correlati con il pacchetto `foo` iniziano, appunto, con il nome "foo" ed hanno le estensioni "preinst", "postinst", ecc. a seconda della funzione. Il file `foo.list` nella stessa directory elenca tutti i files installati con il pacchetto `foo`. (Notate che la localizzazione di questi files è propria di dpkg; non fateci affidamento.)

2.2.6 Priorità

Ad ogni pacchetto viene assegnata una *priorità* dai responsabili della distribuzione, come aiuto al sistema di gestione dei pacchetti. Le priorità sono:

- **Required**: pacchetti necessari al corretto funzionamento del sistema.

  Comprende tutti gli strumenti necessari alla riparazione di difetti di sistema. Questi pacchetti non devono essere rimossi, pena la completa inutilizzabilità del sistema, probabilmente nemmeno con dpkg si riuscirebbe a mettere le cose a posto. I sistemi con solo i pacchetti Required probabilmente sarebbero inutilizzabili, ma hanno abbastanza funzionalità per permettere all'amministratore di sistema di fare un boot ed installare altri programmi.

- **Important** pacchetti che si ritrovano probabilmente su qualsiasi sistema Unix o correlato.

  Altri pacchetti necessari ad un corretto funzionamento del sistema, senza i quali non sarebbe utilizzabile. Tra questi *NON* sono inclusi Emacs o X11 o TeX o qualsiasi altra grossa applicazione. Qui si parla di pacchetti che costituiscono l'infrastruttura di base.

- **Standard** pacchetti comuni su qualsiasi sistema Linux, compreso un sistema ragionevolmente piccolo ma nemmeno troppo limitato all'interfaccia a carattere.

  Questo è ciò che viene installato di base se l'utente non seleziona altro. Non include grosse applicazioni, però include Emacs (più un pezzo di infrastruttura che un'applicazione) ed un ragionevole sottogruppo di TeX e LaTeX (se è possibile senza X).

- **Optional** pacchetti che comprendono tutto quello di cui potete aver voglia di installare senza nemmeno sapere che cosa è, o se non avete delle necessità particolari.

  Comprende X11, una distribuzione completa di TeX e molte applicazioni.

- **Extra**: pacchetti che o entrano in conflitto con altri di priorità più alta, probabilmente utili se già sapete a che servono, oppure hanno requisiti speciali che li rendono non consoni come "Optional".

2.2.7 Pacchetti Virtuali

Il termine pacchetto virtuale è un termine generico che si applica a tutti i pacchetti di un gruppo che provvede alla medesima funzione. Per esempio, i programmi `tin` e `trn` sono entrambi dei newsreader, in grado di soddisfare qualsiasi dipendenza di un programma che richieda un newsreader su un sistema, al fine di funzionare correttamente. Entrambi, quindi, si dice che provvedano il "pacchetto virtuale" definito `news-reader`.

Allo stesso modo `exim` e `sendmail` forniscono entrambi la funzionalità di un agente di trasporto posta (mail transport agent). Entrambi, quindi, provvedono al pacchetto virtuale "mail transport agent". Se uno dei due è installato, qualsiasi programma che dipenda dall'installazione di un `mail-transport-agent` vedrà le proprie dipendenze soddisfatte dall'esistenza di questo pacchetto virtuale.

La Debian ha un meccanismo tale che, se più di un pacchetto che fornisce lo stesso pacchetto virtuale è installato, l'amministratore di sistema è in grado di sceglierne uno come pacchetto preferito. Il comando che viene chiamato in causa è`update-alternatives` e verrà descritto in dettaglio oltre, in 'Comandi alternativi' a pagina 64.

2.2.8 Dipendenze

Il sistema dei pacchetti Debian ha una serie di "dipendenze" che sono pensate per indicare (con un singolo termine) il livello di indipendenza di un dato Programma A a cui può operare, indipendentemente dalla esistenza di un Programma B su un dato sistema:

- Il Pacchetto A *dipende* dal Pacchetto B se B deve essere assolutamente installato per eseguire A. In alcuni casi, esso no dipende solo da B, ma da una sua specifica versione. In tal caso la dipendenza dalla versione rappresenta un limite inferiore, nel senso che A dipende da qualsiasi versione di B più recente di quella specificata.

- Il Pacchetto A *raccomanda* il B, se il responsabile del pacchetto giudica che la maggior parte degli utenti non vorrebbe A senza le funzioni provviste anche da B.

- Il Pacchetto A *suggerisce* B se B contiene files correlati (e che talvolta migliorano) alle funzioni di A.

- Il Pacchetto A *è in conflitto* con B quando A non è in grado di funzionare se B è installato nel sistema. Molto spesso i conflitti si hanno quando A contiene dei files che rappresentano dei miglioramenti di quelli in B. Spesso "è in conflitto" è combinato con "sostituisce".

- Il Pacchetto A *sostituisce* B quando i files installati da B vengono rimossi e (in alcuni casi) sovrascritti da quelli in A.

- Il Pacchetto A *provvede* B quando tutti i files e le funzioni di B vengono incorporate da A. Questo meccanismo permette agli utenti con scarso spazio sul disco rigido di avere solo la parte del pacchetto A che a loro serve realmente.

Informazioni più dettagliate possono essere trovate nel manuale di Packaging ed in quello di Policy.

Bisogna ricordare che `dselect` ha un controllo molto più raffinato sui pacchetti contrassegnati da *raccomanda* e *suggerisce* rispetto ad `apt-get`, che prende semplicemente tutti i pacchetti specificati da *dipende* e lascia quelli indicati da *raccomanda* e *suggerisce*. Entrambi i programmi nelle forme più moderne utilizzano come back-end APT.

2.2.9 Cosa significa Pre-Depends

"Pre-Depends" è una dipendenza speciale. Con la maggior parte dei pacchetti `dpkg` ne spacchetterà il file di archivio (ovvero il file `.deb`), indipendentemente dal fatto che i files da cui dipendono sono o no sul sistema. Semplificando, spacchettare il file vuol dire che `dpkg` ne estrarrà i file da installare e li metterà al loro posto. Se quale determinato pacchetto *dipende* dall'esistenza di altri pacchetti nel sistema,`dpkg` si rifiuterà di completare l'installazione (eseguendo l'azione "configura"), finchè non saranno installati gli altri pacchetti.

Per alcuni pacchetti, tuttavia, `dpkg` si rifiuterà persino di spacchettarli finchè certe dipendenze no vengono risolte. Tali pacchetti "Pre-dipendono" (Pre-depend) dalla presenza di altri pacchetti. Il progetto Debian previde questo meccanismo per supportare un aggiornamento sicuro di sistemi dal formato `a.out` al formato `ELF`, dove *l'ordine* in cui i pacchetti venivano estratti risultava

critico. Esistono altre situazioni di aggiornamenti estesi in cui questo metodo si rivela utile, tipo pacchetti con priorità richiesta e dipendenza da LibC.

Come sopra, informazioni più dettagliate al riguardo possono essere reperite nel manuale di Packaging.

2.2.10 Lo stato dei pacchetti

Lo stato di un pacchetto può essere "sconosciuto", "installa", "rimuovi", "elimina" o "mantieni". Queste etichette "voglio", indicano il volere dell'utente riguardo ad un pacchetto (come indicato dalle azioni dell'utente nella sezione "Scegli" di `dselect` o dal richiamo diretto dell'utente di `dpkg`).

Il loro significato è il seguente:

- sconosciuto - l'utente non ha mai indicato se vuole il pacchetto
- installa - l'utente vuole il pacchetto installato od aggiornato
- rimuovi - l'utente vuole che il pacchetto sia rimosso, ma non i suoi file di configurazione.
- elimina - l'utente vuole il pacchetto completamente rimosso, compresi i file di configurazione.
- mantieni - l'utente non vuole che il pacchetto sia processato, ovvero vuole mantenere la versione attuale con lo stato corrente, qualunque essi siano.

2.2.11 Evitare l'aggiornamento dei pacchetti

Esistono due modi per evitare l'aggiornamento di un pacchetto, tramite "dpkg" o, in woody, tramite APT.

Con dpkg, dovete solo esportare la lista dei pacchetti selezionati con:

```
dpkg --get-selections > selections.txt
```

Dopodichè modificate il file risultante `selections.txt`, cambiando la riga che contiene il pacchetto da mantenere, tipo `libc6`, da:

```
libc6                                           install
```

a:

```
libc6                                                hold
```

Salvate il file e ricaricatelo nel database di dpkg con:

```
dpkg --set-selections < selections.txt
```

Se conoscete il nome del pacchetto da mantenere, basta eseguire:

```
echo
  libc6 hold | dpkg --set-selections
```

Questo processo evita l'aggiornamento dei pacchetti al momento dell'installazione di ciascun file.

Lo stesso risultato si ottiene tramite dselect. Basta accedere alla schermata [S]cegli, trovare il pacchetto da mantenere nello stato attuale e premere il tasto '=' key (o 'H'). I cambiamenti saranno effettivi non appena lasciata la schermata [S]cegli.

Il sistema APT nella nuova distribuzione woody ha un meccanismo alternativo per mantenere i pacchetti durante il processo di raccolta di un archivio, utilizzando la `Pin-Priority`. Vedere la pagina man `apt_preferences(5)`.

2.2.12 Pacchetti sorgente

I pacchetti sorgente vengono distribuiti in una directory chiamata `source` e possono essere scaricati o manualmente, oppure tramite il comando

```
apt-get source foo
```

(vedere `apt-get(8)` la pagina man su come settare APT all'uopo).

2.2.13 Compilare pacchetti binari dai sorgenti

Per compilare i sorgenti, avete bisogno di tutti i files foo_*.dsc, foo_*.tar.gz e foo_*.diff.gz (nota bene: non esiste nessun .diff.gz per un pacchetto Debian nativo).

Una volta presi, se avete installato il pacchetto `dpkg-dev` il seguente comando:

```
dpkg-source -x foo_version-revision.dsc
```

estrarrà il pacchetto in una directory denominata `foo-version`.

Se si vuole semplicemente compilare il pacchetto, si entra nella directory `foo-version` e si lancia il comando

```
debian/rules build
```

per compilare il programma, poi

```
debian/rules binary
```

da root, per compilare il pacchetto ed infine

```
dpkg -i ../foo_version-revision_arch.deb
```

per installarlo.

2.2.14 Creare nuovi pacchetti Debian

Per maggiori dettagli al riguardo, leggete la "New Maintainers' Guide", reperibile nel pacchetto `maint-guide` oppure presso http://www.debian.org/doc/manuals/maint-guide/.

2.3 Aggiornare un sistema Debian

Uno degli scopi della Debian è di fornire un sentiero solido di aggiornamento ed un processo sicuro (sempre di aggiornamento), e si fa sempre del proprio meglio per rendere la nuova versione facilmente aggiornabile dalla precedente. Nel caso di qualcosa di importante da aggiungere al processo di aggiornamento, i pacchetti avvertiranno gli utenti, spesso provvedendo anche ad una soluzione ai possibili problemi.

Bisogna sempre leggere le Note di Rilascio (Release Notes), documento che descrive i dettagli dei singoli aggiornamenti, che viene sempre inserito in tutti i CD Debian, comunque disponibile sul web presso http://www.debian.org/releases/stable/releasenotes oppure presso http://www.debian.org/releases/testing/releasenotes.

Una guida pratica viene fornita in 'Gestione dei pacchetti in Debian' a pagina 53. Questa sezione si occupa dei dettagli fondamentali.

2.3.1 Metodi per aggiornare un sistema Debian

Si può sempre fare un ftp anonimo od un wget ad un archivio Debian e sbirciare nelle directory finchè si trova il file desiderato, scaricarlo ed infine installarlo con `dpkg`. Notate che `dpkg` installerà i file aggiornati al loro posto, persino su un sistema che sta normalmente girando. Talvolta un pacchetto da aggiornare richiederà l'installazione di un altro pacchetto aggiornato, in tal caso l'installazione fallirà finchè/a meno che l'altro pacchetto venga installato.

Molti trovano che un approccio del genere sia troppo dispendioso in termini di tempo, dato che la Debian si evolve molto velocemente – tipicamente una dozzina o più di nuovi pacchetti vengono caricati ogni settimana. Questo numero diventa ben più grande in prossimità di un rilascio di una nuova versione. Per trattare con questa massa di pacchetti, molte persone preferiscono utilizzare programmi automatizzati. Per questo scopo, molti strumenti di gestione dei pacchetti sono disponibili.

2.3.2 Panoramica degli strumenti di gestione dei pacchetti

Il sistema di gestione dei pacchetti Debian è focalizzato su due punti principali, la manipolazione del pacchetto stesso, ed il suo recupero da un archivio Debian.`dpkg` è per il primo punto, `APT` e `dselect` per il secondo.

2.3.3 dpkg

E' il programma principale per la manipolazione dei pacchetti. Per ulteriori informazioni, leggere la pagina man `dpkg(8)`.

`dpkg` è fornito con parecchi programmi supplementari di base.

- dpkg-deb: Manipola i files `.deb`. `dpkg-deb(1)`
- dpkg-ftp: Vecchio comando per il recupero dei pacchetti. `dpkg-ftp(1)`
- dpkg-mountable: Vecchio comando per il recupero dei pacchetti. `dpkg-mountable(1)`
- dpkg-split: Divide grossi pacchetti in files più piccoli. `dpkg-split(1)`

`dpkg-ftp` e `dpkg-mountable` sono stati resi obsoleti dall'introduzione del sistema APT.

2.3.4 APT

APT è un'interfaccia più avanzata per il sistema di gestione dei pacchetti. `apt-get`, `apt-cache` e `apt-cdrom` sono gli strumenti da riga di comando per maneggiare i pacchetti. Funzionano anche come programmi backend per l'utente di altri strumenti, come `dselect` ed `aptitude`.

Per maggiori informazioni, installare `apt` e leggere `apt-get(8)`, `apt-cache(8)`, `apt-cdrom(8)`, `apt.conf(5)`, `sources.list(5)`, `apt_preferences(5)` (woody), e `/usr/share/doc/apt/guide.html/index.html`.

Esistono fonti di informazione alternative, come APT HOWTO (http://www.debian.org/doc/manuals/apt-howto/). Può essere installato tramite `apt-howto` in `/usr/share/doc/apt-howto/en/apt-howto-en.html/index.html`.

`apt-get upgrade` e `apt-get dist-upgrade` hanno la tendenza a prendere tutti i pacchetti elencati sotto "Dipende". Per evitare ciò, usate `dselect`.

2.3.5 dselect

Questo programma rappresenta un'interfaccia utente basata su menu al sistema di gestione dei pacchetti. E' particolarmente utile per prime installazioni ed aggiornamenti su larga scala.

Per ulteriori informazioni, installare `install-doc` e leggere `/usr/share/doc/install-doc/dselect-beginner.en.html` oppure dselect Documentation for Beginners (http://www.debian.org/releases/woody/i386/dselect-beginner).

2.3.6 Aggiornare un sistema in funzione

Il kernel (file system) in debian supporta la sostituzione dei files anche mentre sono in uso.

Viene anche fornito un programma chiamato `start-stop-daemon` che viene impiegato per lanciare i demoni al boot o per fermarli al cambiamento di runlevel del kernel (da multi-utente a singolo, o allo spegnimento della macchina, per esempio). Lo stesso programma viene usato dagli scripts di installazione quando un nuovo pacchetto che contiene un demone viene installato, per fermare i demoni in funzione, e rilanciarli al momento giusto.

E utile notare che il sistema Debian non ha bisogno della modalità singolo utente per aggiornare un sistema in funzione.

2.3.7 File .deb scaricati e tenuti in cache

Se avete scaricato i files nel vostro disco rigido (cosa assolutamente non necessaria, vedere sopra per la descrizione di dpkg-ftp), dopo l'installazione dei pacchetti potete rimuoverli dal vostro sistema.

Se si usa APT, i file vengono tenuti nella directory `/var/cache/apt/archives/`. Potete cancellarli dopo l'installazione, oppure copiarli sulla stessa directory `/var/cache/apt/archives/` di un'altra macchina, per evitare un nuovo download durante la successiva installazione.

2.3.8 Tenere una registrazione dell'aggiornamento

`dpkg` mantiene una registrazione dei pacchetti scompattati, configurati, rimossi e/o eliminati, ma (al momento) non tiene nessuna registrazione dell'attività scritta su terminale durante tali manipolazioni.

Il metodo più semplice per aggirare questo impedimento è di lanciare una qualsiasi sessione di `dpkg/dselect/apt-get` all'interno del programma `script(1)`.

2.4 La sequenza di boot della Debian

2.4.1 `init`

Come ogni buon appartenente alla famiglia degli Unix, Debian esegue il boot eseguendo il programma `init`. Il file di configurazione di `init` (che è `/etc/inittab`) specifica che il primo script da eseguire deve essere `/etc/init.d/rcS`. Questo script controlla e monta i filesystems, carica i moduli, lancia i servizi di rete, imposta l'orologio, esegue altre inizializzazioni e poi lancia tutti gli altri script (tranne quelli con '.' nel filename) localizzati in `/etc/rc.boot/`. Tutti gli script in quest'ultima directory sono riservati all'amministratore di sistema, ed il loro utilizzo nei pacchetti è sconsigliato.

2.4.2 Runlevels

Dopo il completamento del processo di boot, `init` esegue tutti gli script contenuti nella directory specificata dal runlevel di default (dato dalla riga per `id` in `/etc/inittab`). Come la maggior parte degli Unix compatibili con il System V Linux ha 7 runlevels:

- 0 (ferma il sistema),
- 1 (modalità singolo utente),
- 2 a 5 (varie modalità multiutente) e
- 6 (riavvia il sistema).

I sistemi Debian hanno id=2, che indica che il runlevel di default sarà il '2' quando si entra in modalità multiutente e verranno lanciati gli scripts localizzati in `/etc/rc2.d/`.

Di fatto gli scripts localizzati in qualsiasi directory denominata `/etc/rcN.d/` sono semplici link simbolici che si riferiscono a scripts localizzati in `/etc/init.d/`. Tuttavia, i *nomi* dei files in ciascuna directory `/etc/rcN.d/` sono selezionati in modo da indicare il *modo* in cui gli scripts in `/etc/init.d/` saranno lanciati. Entrando nello specifico, prima di entrare in qualsiasi runlevel tutti gli script che iniziano con 'K' vengono lanciati; questi scripts chiudono (uccidono) i servizi. Poi vengono lanciati tutti quelli che iniziano per 'S', che fanno partire i servizi. Il numero a due cifre che segue la lettera 'K' o 'S' indica l'ordine nel quale lo script verrà lanciato. Script con numeri più bassi vengono eseguiti prima.

Questo approccio funziona perchè gli scripts contenuti in `/etc/init.d/` accettano un argomento che può essere 'start', 'stop', 'reload', 'restart' o 'force-reload' eseguendo poi il compito indicato dallo specifico argomento. Questi scripts possono essere utilizzato anche dopo il boot del sistema per controllare svariati processi.

Per esempio, lanciato con l'argomento 'reload', il comando

```
/etc/init.d/sendmail reload
```

manda al demone sendmail un segnale di rileggere il proprio file di configurazione.

2.4.3 Personalizzare il processo di boot

Pare che Debian non usi `rc.local` per personalizzare il processo di boot; quali altre funzioni vengono fornite?

Supponiamo che un sistema debba lanciare lo script `foo` all'avvio, o all'ingresso di uno specifico (System V) runlevel. In tal caso l'amministratore di sistema deve:

- Inserire lo script `foo` nella directory `/etc/init.d/`.
- Lanciare il comando Debian `update-rc.d` con gli argomenti appropriati, per impostare i collegamenti (specificati da riga di comando) fra le directory rc?.d e `/etc/init.d/foo`. Qui '?' è un numero compreso fra 0 e 6 e corrisponde a ciascun runlevel.
- Riavviare il sistema.

Il comando `update-rc.d` imposterà i collegamenti fra i files delle directory rc?.d e lo script contenuto in `/etc/init.d/`. Ogni collegamento inizia con una 'S' o una 'K', seguite da un numero, seguiti dal nome dello script. Gli scripts che iniziano con 'S' in `/etc/rcN.d/` verranno eseguiti quando si entra nel `N`. Quelli che iniziano con 'K' verranno eseguiti lasciando il runlevel `N`.

Si può, per esempio, lanciare lo script `foo` al boot mettendolo `/etc/init.d/` ed impostando i collegamenti con `update-rc.d foo defaults 19`. L'argomento 'defaults' fa riferimento ai runlevels di defaults, che sono quelli da 2 a 5. L'argomento '19' assicura che lo script `foo` verrà chiamato prima di qualsiasi altro script con il numero 20 o maggiore.

2.5 Supportare le differenze

Debian offre parecchie opportunità per soddisfare le esigenze (e i desideri) degli amministratori di sistema, senza per questo renderlo inutilizzabile.

- `dpkg-divert`, vedere 'dpkg-divert' a pagina 63.
- `equivs`, vedere 'Pacchetto `equivs`' a pagina 63.
- `update-alternative`, vedere 'Comandi alternativi' a pagina 64.
- `make-kpkg` può accettare svariati boot loaders. Vedere `make-kpkg(1)`.

Tutti i files in `/usr/local/` appartengono all'amministratore di sistema e debian non li toccherà. Gran parte (o tutti) i files in `/etc` sono `conffiles` e debian non li sovrascriverà in caso di aggiornamento a meno che l'amministratore non lo richieda espressamente.

2.6 Locale non in inglese

Le regole culturali e di linguaggio vengono supportate dalla tecnologia di `locale`. Vedere `locale(7)`.

Un locale rappresenta una serie di regole di linguaggio e culturali. Queste coprono aspetti che vanno dal linguaggio dei messaggi di sistema, ai set di caratteri, convenzioni di pronuncia, ecc. Un programma deve essere in grado di determinare il proprio locale e di agire di conseguenza, per essere trasportabile a culture differenti.

Le seguenti variabili di ambiente vengono valutate in questo modo, per fornire dei valori di locale ai programmi:

1. LANGUAGE: Questa variabile consiste in una lista di nomi di locale, separati da una virgola, in ordine di priorità. Usata solo se il locale POSIX è impostato ad un valore diverso da "C" [questo in woody; la versione in potato ha sempre una priorità maggiore sul locale POSIX]. (estensione GNU)
2. LC_ALL: Se non-zero, il valore è usato per tutte le categorie di locale. (POSIX.1) In genere "" (zero).

3. LC_*: Se non-zero, il valore è utilizzato per la categoria corrispondente (POSIX.1). In genere "C". LC_* le varibili sono:
   - LC_CTYPE: Classificazione dei caratteri e conversione dell'altezza.
   - LC_COLLATE: Ordine di collazione.
   - LC_TIME: Formati data ed ora.
   - LC_NUMERIC: Formati numerici non monetari.
   - LC_MONETARY: Formati monetari.
   - LC_MESSAGES: Formati dei messaggi informativi e diagnostici e delle risposte interattive.
   - LC_PAPER: Formato foglio.
   - LC_NAME: Formati dei nomi.
   - LC_ADDRESS: Formati degli indirizzi e delle informazioni dei luoghi.
   - LC_TELEPHONE: Formati dei numeri telefonici.
   - LC_MEASUREMENT: Unità di misura (Decimale od altro).
   - LC_IDENTIFICATION: Metadati sulle informazioni di locale.

4. LANG: Se non-zero e se LC_ALL è indefinito, il valore viene utilizzato per tutte le categorie LC_* di locale con valori indefiniti. (POSIX.1) In genere "C".

Notate che alcune applicazioni (tipo Netscape 4) ignorano le impostazioni LC_*.

Introduction to i18n (`http://www.debian.org/doc/manuals/intro-i18n/`) è indirizzato agli sviluppatori, però è utile anche per gli amministratori.

- debian è distribuita con le keymaps per più di venti tastiere e con delle utilità (contenute nel pacchetto `kbd`) per installare, visualizzare e modificare le configurazioni.

  All'installazione viene richiesto all'utente di specificare la tastiera che utilizzerà.

- La gran parte del software che accompagna la debian supporta i caratteri non-US-ASCII usati da altre lingue latine (tipo. ISO-8859-1 o ISO-8859-2), ed un certo numero di programmi supporta linguaggi multi-byte tipo il giapponese o il cinese.

- Attualmente il supporto per le pagine man in tedesco, spagnolo, finlandese, francese, ungherese, italiano, giapponese, coreano e polacco viene fornito tramite i pacchetti `manpages-LANG` (dove LANG è il codice nazionalità ISO a due lettere). Per accedere alle pagine man in un linguaggio diverso dall'inglese, l'utente deve impostare la variabile di shell LC_MESSAGES con la stringa appropriata.

  Per esempio, per le pagine man in italiano, LC_MESSAGES deve essere impostata su 'italian'. Il programma `man` cercherà allora le pagine man in italiano, sotto `/usr/share/man/it/`.

2.7 Debian ed il kernel

2.7.1 Compilare un kernel non debian

Bisogna comprendere la politica debian degli headers.

Le librerie C Debian sono compilate con le versioni *stabili* più recenti degli headers del **kernel**.

Ad esempio, le versione Debian-1.2 usava la versione 5.4.13 degli headers. Questa pratica è in contrasto con i pacchetti sorgente del kernel distribuiti in tutti gli archivi Linux FTP, pacchetti che usano versioni persino più recenti degli headers. Gli headers distribuiti con i sorgenti del kernel sono localizzati in `/usr/include/linux/include/`.

Se avete bisogno di compilare un programma con headers più recenti di quelli di quelli forniti da `libc6-dev`, quando compilate dovete aggiungere alla riga di comando `-I/usr/src/linux/include/`. Un problema del genere è uscito, per esempio, quando si è creato il pacchetto del demone automounter (`amd`). Quando i nuovi kernels cambiavano alcune istruzioni relative al NFS, `amd` aveva necessità di esserne al corrente. Ciò ha richiesto l'inclusione degli headers più recenti.

2.7.2 Gli strumenti per compilare un kernel personalizzato.

Gli utenti che desiderano (o devono) compilare un kernel personalizzato, sono incoraggiati a scaricare il pacchetto `kernel-package`. Il pacchetto contiene lo script per compilare il pacchetto del kernel e fornisce le capacità di creare un pacchetto Debian kernel-image, semplicemente lanciando il comando

```
make-kpkg kernel_image
```

dalla directory principale del kernel sorgente. L'aiuto è disponibile dando il comando

```
make-kpkg --help
```

o tramite la pagina man `make-kpkg(8)`.

L'utente deve scaricarsi a parte il sorgente per il kernel, sia esso il più recente o quello di scelta, dall'archivio Linux preferito, a meno che un pacchetto kernel-source-version non sia disponibile (dove 'version' sta per la versione del kernel). Vedere 'Ricompilare il kernel' a pagina 67.

Le istruzioni dettagliate per usare il pacchetto `kernel-package` sono fornite nel file `/usr/doc/kernel-package/`

2.7.3 Boot loaders alternativi

Per utilizzare boot loaders alternativi a LILO, tipo `grub` o `loadlin`, copiate il kernel compilato `bzimage` in un'altra locazione (tipo /boot/grub od una partizione `MS-DOS`).

2.7.4 Boot floppy personalizzato

Questo compito è fortemente aiutato dal pacchetto Debian `boot-floppies`, reperibile normalmente nella sezione `admin` dell'archivio FTP Debian. Gli script di shell di questo pacchetto producono dei boot floppies nel formato `SYSLINUX`. Questi sono floppy formattati `MS-DOS`i cui master boot records sono stati modificati in maniera tale da fare il boot di Linux (o di qualsiasi altro S.O. sia stato definito nel file syslinux.cfg del floppy) direttamente. Altri script del pacchetto producono dei dischi root di emergenza e possono persino riprodurre i dischi base.

Maggiori informazioni in `/usr/doc/boot-floppies/README` dopo l'installazione del pacchetto `boot-floppies`.

2.7.5 Funzioni speciali per trattare con i moduli

Il pacchetto Debian `modconf` fornisce uno script di shell (`/usr/sbin/modconf`) che può essere utilizzato per personalizzare la configurazione dei moduli. Lo script presenta un'interfaccia a menu, chiedendo all'utente particolari circa i device drivers caricabili presenti sul proprio sistema. La risposte vengono utilizzate per personalizzare il file `/etc/modules.conf` (che elenca alias ed altri argomenti che devono essere utilizzati insieme ai vari moduli), tramite i files in `/etc/modutils/`,e `/etc/modules` (che elencano i moduli che devono essere caricati al boot).

Così come i (nuovi) files Configure.help ora disponibili per aiutare nella compilazione di kernels personalizzati, il pacchetto modconf arriva con tutta una serie di file di aiuto (in `/usr/lib/modules_help/`) che forniscono informazioni dettagliate sugli argomenti appropriati da dare a ciascun modulo.

2.7.6 Disinstallare un vecchio pacchetto kernel

Si, lo script `kernel-image-NNN.prerm` controlla se il kernel attualmente in uso è lo stesso che state tentando di disinstallare. Perciò potete rimuovere pacchetti kernel che non volete più tramite il comando:

```
dpkg --purge --force-remove-essential kernel-image-NNN
```

(sostituite "NNN" con la versione ed il numero di revisione del vostro kernel, naturalmente)

Capitolo 3

Installazione di Debian

La documentazione ufficiale sull'installazione di Debian è localizzata presso http://www.debian.org/releases/stable/ e http://www.debian.org/releases/stable/installmanual.

Per le versioni in fase di sviluppo è localizzata presso http://www.debian.org/releases/testing/ e http://www.debian.org/releases/testing/installmanual.

Sebbene "La guida Debian" data dai giorni del rilascio della distribuzione potato, gran parte del suo contenuto è indirizzato all'installazione di una Debian Woody (3.0r0).

3.1 Installazione di Linux

Nell'ottica di ridurre al minimo i rischi connessi con l'uso dei pacchetti da testing ed unstable, è buona pratica avere un sistema con un boot alternativo, in un sistema piccolo e stabile.

3.1.1 Le basi della compatibilità hardware

Linux è compatibile con la maggior parte dei componenti per PC esistenti. Può essere installato praticamente in ogni configurazione. Per quanto mi riguarda, l'installazione è stata semplice come per Window$ 95/98/Me. La lista di componenti compatibili cresce di giorno in giorno.

Se avete un laptop, andate su: Linux on Laptops (http://www.linux-laptop.net/) per controllare la situazione compatibilità.

La mia raccomandazione per i componenti di un desktop è, invece, "Siate conservativi."

- SCSI invece di IDE per il lavoro, IDE/ATAPI HD per uso privato.
- IDE/ATAPI CD-ROM (o CD-RW).

- PCI invece di ISA. In particolare per le NIC.
- Usate NIC economiche. Tulip per PCI, NE2000 per ISA sono più che buone.
- Evitare PCMCIA (notebook) se siete alla prima installazione.
- Evitate le tastiere, mouse, ecc. USB, se non volete problemi.

Per macchine lente, bypassarne il disco rigido ed interfacciarle con macchine più veloci per eseguire l'installazione di Linux può essere una buona idea.

3.1.2 Definire i componenti del PC — il chipset

Durante l'installazione vi verranno richieste informazioni sull'hardware o sui chipset, informazioni non sempre facili da reperire. Ecco i miei consigli:

1. Aprite il PC e guardate dentro.
2. Scrivete i numeri presenti sui grossi chip, sulle schede grafiche, sulle schede di rete, sui chip situati vicino alle porte seriali ed alle porte IDE.
3. Scrivete i nomi scritti sul dorso delle schede PCI ed ISA.

3.1.3 Definire i componenti del PC — Debian

Dare i comandi seguenti all'interno di un sistema Linux già installato, fornisce qualche idea sull'hardware presente e sulla sua configurazione.

```
$ /sbin/lspci -v |less
$ less /proc/pci
$ less /proc/interrupts
$ less /proc/ioports
```

3.1.4 Definire i componenti del PC — altri S.O.

Una ulteriore fonte di informazione è rappresentata dagli altri sistemi operativi.

Installate una distribuzione di Linux commerciale. I sistemi di riconoscimento dell'hardware tendono ad essere migliori della Debian. E' una cosa che potrebbe cambiare con l'arrivo di woody.

Installate Windows. La configurazione attuale del PC può essere ottenuta cliccando con il pulsante destro del mouse su "Risorse del Computer" e poi "Proprietà", ottenendo informazioni tipo IRQ, indirizzi di porta I/O, DMA. Alcune vecchie schede ISA potrebbero aver bisogno di essere configurate sotto DOS, ed utilizzate di conseguenza.

3.1.5 Il mito Lilo

Lilo è limitato ai primi 1024 cilindri. — SBAGLIATO !

Il nuovo lilo, usato nella Debian potato ha il supporto lba32. Se il BIOS o la scheda madre sono nuovi e possono supportare lba32, lilo dovrebbe essere in grado di caricarsi oltre il vecchio limite dei 1024 cilindri.

Se avete tenuto il vostro vecchio lilo.conf, vi basterà aggiungere una riga con "lba32" da qualche parte vicino all'inizio del file.

3.1.6 Scegliere i boot floppies

Il set di dischi denominato IDEPCI è il migliore se state installando su un desktop. Il kernel IDEPCI sul dischetto boot omonimo abilita le schede di rete PCI, permettendovi di ottenere praticamente tutto dalla rete. Servono solo due floppy disk (boot e root).

Per sistemi particolari, potete creare un disco di salvataggio fatto apposta, sostituendo l'immagine del kernel denominata "linux" sul disco di salvataggio Debian con un'altra immagine compilata apposta per la suddetta macchina. I dettagli dell'operazione li trovate sul file readme.txt, sul dischetto di salvataggio. Il floppy è formattato con il filesystem MSDOS, per cui potete utilizzare qualunque sistema per leggerlo e modificarlo. Ciò dovrebbe rendere la vita più semplice a coloro con particolari schede di rete, ecc..

Per schede di rete PCMCIA, configuratele con il setup specifico per PCMCIA e non con il setup generico di rete.

3.1.7 Installazione

Le note di rilascio ufficiali per l'aggiornamento sono reperibili presso `http://www.debian.org/releases/stable/releasenotes` e `http://www.debian.org/releases/testing/releasenotes` (in fase di completamento).

Alcuni consigli veloci per installare potato:

1. Preparate i dischetti rescue/root(/driver). (IDEPCI per il desktop)
2. Fate il boot con i dischetti rescue e root.
3. Fdisk/fsck/mount swap, root, tmp, var, home, usr, ecc.. (nessun supporto 2.0)
4. Installate il S.O.. (Se avete IDEPCI, installabile via rete)
5. Se utilizzate il floppy con il kernel standard, installate ora i drivers.

6. Configurate i drivers. (Non ci sono opzioni se IDE)
7. Installate il sistema base dalla rete o dal floppy. (base2_2.tgz) [WOODY: potrebbe non essere qui]
8. Configurate il sistema base.
9. Non installate lilo nel MBR, ed utilizzate un un mbr multi-boot.
10. Fate il reboot. (non prendetevi la briga di creare un disco boot)
11. Sì MD5, sì shadow, configurate anche gli user account
12. Installate le funzioni avanzate (dselect **)
13. Scegliete il minimo indispensabile ai vostri bisogni. Evitate emacs, nvi, tex, telnet, talk(d), includete mc, vim, kernel-image-2.2.18pre21 (kernel completo se avete usato un disco IDE per l'installazione), ...
14. Installate (scaricate tutto...)
15. Rispondete si (= y) a tutte le domande di configurazione (sostituisce l'attuale)
16. exim: scegliete 2 se siete dietro un firewall, 1 per una macchina per Internet.

Per ulteriori informazioni su dselect, vedere 'Dselect - configurazione generale' a pagina 61.

3.1.8 Quale host ed IP usare per LAN

Esempio di configurazione di una LAN (C subnet: 192.168.1.0/24):

```
Internet
   |
   +--- Un ISP esterno fornisce il servizio POP (raggiunto da fetchmail)
   |
  Un punto di accesso dell'ISP fornisce i servizi DHCP e SMTP
   |                    :
  Modem             (Dial-up)
   |                    :
Porta esterna della macchina gateway della LAN: eth0 (IP dato dal DHCP dell'ISP)
 usa un vecchio notebook (IBM Thinkpad, 486 DX2 50 MHz, 20 MB RAM)
 gira Linux con kernel 2.4 e ext3 fs.
 gira il pacchetto "ipmasq" (con patch più sicuro, NAT e firewall)
 gira il pacchetto "dhcp-client" configurato per eth0 (scavalca le impostazioni DNS
```

```
 gira il pacchetto "dhcp" configurato per eth1
 gira "exim" come smarthost (modo 2)
 gira "fetchmail" con un intervallo lungo (fall back)
 gira "bind" come cache name server per Internet dalla LAN
           come name server autoritativo per il dominio LAN dalla LAN
 gira "ssh" sulle porte 22 e 8080 (connessione da ovunque)
 gira "squid" come cache server per l'archivio Debian (per APT)
Porta interna della macchina gateway della LAN: eth1 (IP = 192.168.1.1, fisso)
                          |
         +--- LAN Switch (10 base T) ---+
         |                              |
Alcuni clienti IP fissi sulla LAN     Alcuni clienti DHCP sulla LAN
(IP = 192.168.1.2-127, fisso)    (IP = 192.168.1.128-200, dinamico)
```

Vedere 'Costruire un gateway con Debian' a pagina 101 per i dettagli sulla configurazione del gateway server.

3.1.9 Account utenti

Per avere lo stesso aspetto di "famiglia" tra le varie macchine, i primi accounts che imposto sono sempre gli stessi.

Il primo account che creo è sempre uno del tipo "admin" (uid=1000). Tutta la posta per il root viene reindirizzata lì. Questo account è aggiunto al gruppo staff, al quale viene data una certa quota di privilegi di root tramite il comando sudo. Vedere 'Aggiungere un account utente' a pagina 42 per i dettagli.

3.1.10 Partizionare il disco rigido ed impostare l'NFS

Per limitare i danni in caso di crash del sistema, preferisco tenere partizioni diverse per directory diverse. Per esempio,

```
/          == (/ + /boot + /bin + /sbin)
           == 50MB+
/tmp       == 100MB+
/var       == 100MB+
/home      == 100MB+
/usr       == 700MB+ con X
/usr/local == 100MB
```

Le dimensioni della directory `/usr` dipende strettamente dal tipo di applicazione X-window. `/usr` può essere di soli 300MB per la sola console, di 2-3 GB se si vogliono installare varie applicazioni di Gnome. Quando `/usr` diventa troppo grande, la cura più efficace è spostare `/usr /share/` in una partizione diversa. Con i nuovi kernel 2.4 pre-impacchettati, `/` può avere bisogno di più di 200MB di spazio.

Per esempio, la configurazione attuale della mia macchina che funge da Internet gateway, è la seguente (output del comando `df -h`):

```
Filesystem Size Used Avail Use% Mounted on /dev/hda3 300M 106M 179M 38% /
/dev/hda7 100M 12M 82M 13% /home /dev/hda8 596M 53M 513M 10% /var /dev/hda6
100M 834k 94M 1% /var/lib/cvs /dev/hda9 596M 222M 343M 40% /usr /dev/hda10
596M 130M 436M 23% /var/cache/apt/archives /dev/hda11 1.5G 204M 1.2G 14%
/var/spool/squid
```

Le grosse dimensioni di /var/spool/squid sono per la funzione di proxy e per lo scarico dei pacchetti.

Per dare un'idea, il seguente è l'output di `fdisk -l`.

```
# fdisk -l /dev/hda # commenti

/dev/hda1 1 41 309928+ 6 FAT16 # DOS /dev/hda2 42 84 325080 83 Linux # (non
utilizzata) /dev/hda3 * 85 126 317520 83 Linux # Main /dev/hda4 127 629
3802680 5 Extended /dev/hda5 127 143 128488+ 82 Linux swap /dev/hda6 144 157
105808+ 83 Linux /dev/hda7 158 171 105808+ 83 Linux /dev/hda8 172 253 619888+
83 Linux /dev/hda9 254 335 619888+ 83 Linux /dev/hda10 336 417 619888+ 83
Linux /dev/hda11 418 629 1602688+ 83 Linux
```

Ci sono alcune partizioni inutilizzate. Queste servono per installare una seconda distribuzione di Linux, o come spazio per espandere qualche directory in crescita.

Se avete più di due dischi, create le partizioni di swap su ognuno, per guadagnare il massimo delle prestazioni.

Montare appropriatamente le partizioni avviene mediante il seguente `/etc/fstab`.

```
# /etc/fstab: static file system information.  file system mount point type
#options dump pass /dev/hda3 / ext2 defaults,errors=remount-ro 0 1 /dev/hda5
#none swap sw 0 0 proc /proc proc defaults 0 0 /dev/fd0 /floppy auto
#defaults,user,noauto 0 0 /dev/cdrom /cdrom iso9660 defaults,ro,user,noauto 0
```

```
#0 mantenete le partizioni separate /dev/hda7 /home ext2 rw 0 2 /dev/hda8 /var
#ext2 rw 0 2 /dev/hda6 /var/lib/cvs ext2 rw 0 2 /dev/hda9 /usr ext2 rw 0 2
#/dev/hda10 /var/cache/apt/archives ext2 rw 0 2

# una partizione molto grande come proxy cache /dev/hda11 /var/spool/squid
ext2 rw 0 2

# DOS bootabile di backup /dev/hda1 /mnt/dos vfat rw,noauto 0 0 # Linux
bootabile di backup (non ancora fatto) /dev/hda2 /mnt/linux ext2 rw,noauto 0 0
# # nfs mounts mickey:/ /mnt/mickey nfs ro,noauto,intr 0 0 goofy:/ /mnt/goofy
nfs ro,noauto,intr 0 0 # minnie:/ /mnt/minnie smbfs
ro,soft,intr,credentials={filename} 0 2
```

Io uso qui le opzioni “noauto,intr” combinate con quella di default “hard” per nfs. In tal modo, un processo che si blocca in seguito alla perdita di connessione può essere recuperato mediante control-C.

Usare “rw,auto,soft,intr” per macchine con windows connesse tramite samba (smbfs), può essere una buona idea. (FIXME)

Controllare autofs (FIXME)

Per i floppy, utilizzare “noauto,rw,sync,user,exec” previene la corruzione dei file in caso di rimozione accidentale del disco prima di averlo smontato, però rallenta il processo di scrittura.

Il server esterno nfs (goofy) risiede dietro il firewall (gateway). Dato che sono l’unico ad usarla, ho delle regole di sicurezza sulla LAN molto allentate. Per abilitare l’accesso nfs, il server nfs necessita di aggiungere `/etc/exports` come segue:

```
# /etc/exports: lista di controllo degli accessi per filesystems che possono
essere esportati ai clienti NFS. Vedere exports(5).  / (rw,no_root_squash)
```

Ciò è richiesto per attivare il server nfs, in aggiunta all’installazione e ad attivazione del server/client nfs.

Per semplicità, in genere creo un partizione singola di circa 2GB per installazioni sperimentali e/o secondarie di linux. Opzionalmente condivido le partizioni di swap e /tmp. Per questi scopi lo schema a partizioni multiple è eccessivo. Se vi serve un semplice sistema a console, bastano 500MB.

3.1.11 Linee guida per la memoria DRAM

Di seguito presento alcune (grandi) linee guida per le DRAM.

```
 4 MB: Il minimo necessario a far funzionale il kernel di Linux.  16 MB: Il
minimo per un sistema con console.  32 MB: Il minimo per un semplice sistema
X.  64 MB: Il minimo per X con GNOME/KDE.  128 MB: Il giusto per X con
GNOME/KDE.  256+MB: Perché no? se ve le potete permettere. Le DRAM sono (ora)
economiche.
```

L'opzione di boot mem=4m (o lilo append="mem=4m") vi mostra come il sistema si comporta con soli 4MB di memoria installati. Il parametro di boot per lilo è necessario per sistemi con vecchi BIOS e più di 64MB di memoria.

3.1.12 Lo spazio di swap

Uso la formula seguente:

```
Swap totale = min(1x-2x RAM installata, 128 MB - 1 GB) Ogni swap < 128 MB
Tenetele su dischi diversi.  Se ne ho la necessità, nella parte centrale del
disco rigido.
```

Anche se non ne avete mai bisogno, un pò di spazio swap (128MB) è comunque desiderabile, in modo che un programma che ha un leak di memoria rallenti progressivamente il sistema prima di bloccarlo definitivamente.

3.2 Configurare Bash

Modifico gli script di inizio della shell a mio piacimento:

```
/etc/bash.bashrc Sostituire con copia privata /etc/profile Mantenere la copia
della distribuzione ( \w -> \W) /etc/skel/.bashrc Sostituire con copia
privata /etc/skel/.profile Sostituire con copia privata
/etc/skel/.bash_profile Sostituire con copia privata ~/.bashrc Sostituire con
copia privata per tutti gli accounts ~/.profile Sostituire con copia privata
per tutti gli accounts ~/.bash_profile Sostituire con copia privata per tutti
gli accounts
```

Per i dettagli vedere i miei esempi. (examples/) Mi piacciono i sistemi trasparenti, così imposto l'umask 002 o 022.

PATH viene impostato dai seguenti files di configurazione, in questo ordine.

```
/etc/login.defs - prima del PATH impostato per la shell /etc/profile ( può
chiamare /etc/bash.bashrc) ~/.bash_profile ( può chiamare ~/.bashrc)
```

3.3 Configurare il mouse

In caso di un mouse PS/2 con scheda madre ATX, il flusso del segnale deve essere:

```
mouse -> /dev/psaux -> gpm -> /dev/gpmdata = /dev/mouse -> X
```

Ciò permette alla tastiera/mouse di essere staccati e reinizializzati lanciando nuovamente gpm alla riconnessione. X rimane funzionante.

Per il mouse PS2 Logitech a tre pulsanti, la combinazione delle configurazioni dovrà essere:

```
/etc/gpm.conf /etc/X11/X86Config o X86Config4
============================================================
device=/dev/psaux Section "Pointer" responsiveness= Protocol "IntelliMouse"
repeat_type=ms3 Device "/dev/gpmdata" type=ps2 append=""
------------------------------------------------------------
device=/dev/psaux Section "Pointer" responsiveness= Protocol "IntelliMouse"
repeat_type=raw Device "/dev/gpmdata" type=ps2 append=""
```

Se si usa un normale mouse PS2 a due pulsanti, impostate il protocollo di X a “Microsoft” ed abilitate “Emulate3Buttons”. Per mouse con lo scrolling, potete dare ad X il vero protocollo “IMPS/2”, utilizzando il secondo esempio. Create un softlink /dev/gpmdata –> /dev/mouse per accontentare alcune utilità di configurazione. Per i dettagli vedere i miei esempi. (examples/)

Per alcuni nuovi notebook Toshiba ultrapiatti: Attivare gpm prima di PCMCIA ngllo script di inizializzazione del System-V. Ciò evita che gpm si pianti. Strano, ma vero.

3.4 Configurare NFS

Impostate NFS tramite /etc/exports.

```
# echo "/ *.domainname-for-lan-hosts(rw,no_root_squash,nohide)" \ >>
        /etc/exports
```

Per i dettagli vedere i miei esempi. (examples/)

3.5 Configurare Samba

Impostare Samba in modalità “share” è molto più semplice, dato che crea un drive di share sul modello WfW. E’ comunque molto meglio l’impostazione in modalità “user”.

Samba può essere configurato con `sambaconfig` o vi:

```
$su -c "sambaconfig" $su -c "vi /etc/samba/smb.conf"
```

Per i dettagli vedere i miei esempi. (`examples/`)

L’aggiunta di un nuovo utente al file smbpasswd può essere fatta tramite `smbpasswd`:

```
$su -c "smbpasswd -a nomeutente"
```

Per la migliore compatibilità, usate password criptate.

Livello OS significa quanto segue, maggiore il numero, più alta la priorità come server.

```
0: Samba con attitudini molto lasse 1: Wfw 3.1, Win95, Win98, Win/me?  16: Win
NT WS 3.51 17: Win NT WS 4.0 32: Win NT SVR 3.51 33: Win NT SVR 4.0 255: Samba
con poteri estesi
```

La directory condivisa deve essere leggibile ed eseguibile da un determinato gruppo, e l’utente deve far parte di detto gruppo.

3.6 Configurare la stampante

Installate LPRng al posto dello standard lpr.

```
# apt-get install lprng enscript gs
```

Allo stato attuale (Debian potato2.2r2), printtool e lprngtool non sono in potato. Scaricatene uno da woody/binary-all/admin ed installatelo con “dpkg -i”.

Se una macchina con window$ deve stampare tramite samba, l’accesso può avvenire senza il filtro GS utilizzando il proprio windriver, oppure con il filtro tramite drivers PS.

3.7 Altri consigli di configurazione per l'host

3.7.1 Installate pochi altri pacchetti ed attenetevi alla configurazione di base

Vedere 'Dselect - configurazione generale' a pagina 61. In genere elimino (_): TEX, nvi, ae, lynx, ed aggiungo (+): vim, ssh, lynx-ssl, mc.

Modificate /etc/inittab con CTRL-DEL-ALT=halt per un facile shutdown.

3.7.2 Moduli

Impostateli durante la configurazione dei drivers durante l'installazione. La lista dei moduli è in /etc/modules. Per un controllo manuale, io utilizzo anche lsmod e depmod.

Aggiungo anche poche linee in /etc/modules per gestire l'ip-masquerading (ftp ecc.) per i kernels 2.4. Per i dettagli vedere i miei esempi. (examples/)

3.7.3 Impostazione base del CD-RW

Modificate i files seguenti:

```
/etc/lilo.conf  (append="hdc=ide-scsi", lanciate lilo per attivarlo)
/dev/cdrom      (softlink # cd /dev; ln -sf scd0 cdrom)
/etc/modules    (aggiungere "ide-scsi" and "sg". "sr" a seguire, se
                 necessario.)
```

Vedete 'Masterizzatori-writer' a pagina 88 per i dettagli.

3.7.4 Grosse memorie e spegnimento automatico

Modificate /etc/lilo.conf come segue per impostare i parametri al boot prompt in caso di grosse memorie (per i kernel 2.2) e per lo spegnimento automatico (per apm):

```
append="mem=128M apm=on apm=power-off"
```

Lanciate lilo per installare queste nuove impostazioni. "apm=power-off" è necessario per il SMP-kernel. Si può fare lo stesso direttamente al boot prompt. Vedere 'Altri trucchi per il boot prompt' a pagina 74.

Se apm è compilato come modulo, come avviene automaticamente nei kernels Debian 2.4, lanciate "# insmod apm power_off=1" dopo il boot, oppure impostate /etc/modules con:

```
# echo "apm power_off=1" >>/etc/modules
```

In alternativa, compilare il supporto ACPI raggiunge lo stesso scopo con i kernels più nuovi, e sembra essere più "gentile" con SMP (richiede l'ultimo Mobo). I kernels 2.4 con l'ultimo Mobo dovrebbero riconoscere le grosse memorie correttemente.

```
CONFIG_PM=y
CONFIG_ACPI=y
...
CONFIG_ACPI_BUSMGR=m
CONFIG_ACPI_SYS=m
```

ed aggiungete le seguenti righe a /etc/modules nell'ordine.

```
ospm_busmgr
ospm_system
```

Oppure ricompilate il kernel con tutte le opzioni precedenti su "y". Nel caso dell'ACPI, non serve alcun parametro di boot.

3.7.5 Altre modifiche di configurazione in /etc/*

Aggiunti i files seguenti /etc/cron.deny (mancante, copiato /etc/at.deny)

Capitolo 4

Lezioni di Debian

Questa sezione fornisce delle coordinate di base per orientarsi nel mondo Linux, mirate all'utente inesperto. Se è un pò di tempo che bazzicate con Linux, usatela come ripasso.

4.1 Fonti di informazione

```
# apt-get install doc-debian doc-linux-text apt-howto info \
                  doc-debian-it doc-linux-it manpages-it \
                  man-db manpages less mc
```

Sbirciate nel Debian Documentation Project (DDP) (http://www.debian.org/doc/ddp) che ha i riferimenti più importanti per Debian. Molti di questi documenti li trovate già installati in /usr/share/doc. Date un'occhiata anche in /usr/share/doc-base/, che fornisce indicazioni sulla localizzazione dei documenti sul sistema. Aggiungete al vostro ~/.bash_profile "export CDPATH=.:/usr/share/doc:/usr/src/local" per un accesso più facile alle directories contenenti la documentazione.

Il Linux Documentation Project (LDP) (http://www.linuxdoc.org/) ha i riferimenti più importanti per quel che attiene a Linux. I documenti dell'LDP sono in genere installati sotto /usr/share/doc/HOWTO/.

Con Midnight Commander potete navigare fra i documenti contenuti sulle directories locali o su siti remoti ftp, usando F9. (vedere 'Midnight Commander (MC)' a pagina 44).

4.2 La console in Linux

4.2.1 Login

In un sistema Linux standard esistono 6 pseudo-terminali indipendenti. Potete passare da uno all'altro premendo simultaneamente i tasti `Left-Alt` e `F1 - F6`. Ciascun (pseudo)terminale permette i login indipendenti agli account. L'ambiente multiutente è una grande e avvincente caratteristica di Unix.

E' ritenuta una buona abitudine nell'ambito Unix, accedere al sistema tramite un account regolare per la maggior parte delle operazioni. Devo ammettere che uso l'account superuser (root) molto più di quanto sarebbe realmente necessario, a causa della sua utilità e della mia pigrizia.

Ora come ora, utilizzo un account regolare con i comandi "sudo", "super" o "su -c" per avere un accesso limitato a root.

4.2.2 Aggiungere un account utente

Terminata l'installazione, di solito aggiungo un account utente standard. Se lo username è "penguin", per esempio,

```
# adduser penguin
```

creerà il nuovo account.

Per modificare /etc/group come segue, uso il comando "vigr":

```
src:x:40:admin, debian, ...
staff:x:50:admin
...
```

Guardate i comandi "adduser", "addgroup", "vipw", "vipw -s", "vigr", e "vigr -s" per una corretta configurazione di gruppi ed utenti.

4.2.3 Lo spegnimento del PC

Come qualsiasi moderno sistema operativo, in cui i files vengono tenuti in memoria, Linux necessità di una procedura di arresto appropriata, prima che l'interruttore possa essere spento con sicurezza. In modalità multiutente il comando è il seguente:

```
# shutdown -h now
```

In modalità singolo utente è invece:

```
# poweroff -i -f
```

Attendete finchè non appare la scritta "System halted", poi spegnete il computer. Se è attivo sul BIOS e su Linux l'apm, il sistema si spegnerà da solo. Per i dettagli, vedere 'Grosse memorie e spegnimento automatico' a pagina 39.

4.2.4 L'editing della riga di comando

La shell di default, `bash`, ha la capacità di ricordare i comandi utilizzati. Basta usare il tasto freccia-su per richiamare i comandi precedenti, usando poi i tasti cursore per muoversi al loro interno. Altre combinazioni da ricordare sono:

```
Ctrl-C:                  Termina il programma
Ctrl-D:                  Termina l' input
Ctrl-S:                  Blocca l'output allo schermo
Ctrl-Q:                  Riattiva l'output allo schermo
Ctrl-Alt-Del:            Reboot/arresto del sistema
                         (vedere /etc/inittab)
Click-sn-e-muovi-mouse:  Seleziona e copia negli appunti (gpm)
Ctrl-click-mouse:        Incolla il testo dagli appunti al cursore (gpm)
```

Su una console Linux standard, solo i tasti `Ctrl` e `Alt` sinistri funzionano come ci si aspetta.

4.2.5 Comandi base da ricordare

I seguenti sono i comandi base in Unix:

```
ls, ls -al, ls -d, pwd, cd, cd ~user, cd -, cat /etc/passwd, less,
bg, fg, kill, killall, uname -a, type commandname, sync, netstat,
ping, traceroute, top, vi, ps aux, tar, zcat, grep, ifconfig, ...
```

Imparate il loro significato dando il nome del comando al prompt, oppure dando "man " o "info " più il nome del comando. Molti comandi in Linux mostrano una breve nota informativa se invocati in uno dei seguenti modi:

```
$ nomecomando --help
$ nomecomando -h
```

"whatis *commandname*" fornisce un riassunto di una riga per qualsiasi comando del sistema per il quale esiste una voce di manuale.

4.2.6 Il sistema X Window

Per lanciare il sistema X Window da console il comando è:

```
# exec startx
```

Cliccando col pulsante destro del mouse la finestra principale (root window) si apre il menu per le selezioni principali.

4.2.7 Altre combinazioni di tasti

Alcune combinazioni da ricordare per la console:

```
Alt-F1 fino a F6:      Passa da un terminale all'altro
Ctrl-Alt-F1 fino a F6: Passa ad altri terminali
                       (da un X-window, DOSEMU, ecc.)
Alt-F7:                Torna ad X
Ctrl-Alt-meno:         Modifica la risoluzione dello schermo in X
Ctrl-Alt-più:          Modifica la risoluzione in X dalla parte opposta
Ctrl-Alt-Backspace:    Termina X-windows
Alt-X, Alt-C, Alt-V:   Le normali combinazioni Windows/Mac per Taglia,
        Copia, Incolla con Ctrl- in alcuni programmi tipo Netscape
        Composer vengono s ostituite da Alt-.
```

4.3 Midnight Commander (MC)

Midnight Commander (MC) può essere considerato come uno di quei coltellini multiuso, prodotto da GNU, per la console ed altri terminali.

4.3.1 Installazione di MC

```
# apt-get install mc
```

Dopodichè aggiungete a ~/.bashrc (oppure /etc/bash.bashrc, chiamato da .bashrc) la seguente funzione.

```
mc ()
{
  mkdir -p ~/.mc/tmp 2> /dev/null
  chmod 700 ~/.mc/tmp
  MC=~/.mc/tmp/mc-$$
  /usr/bin/mc -P "$@" > "$MC"
  cd "$(cat $MC)"
  rm -f "$MC"
  unset MC;
}
```

Ciò permette a MC di cambiare la directory di lavoro all'uscita.

Se siete in un terminale, tipo kon e Kterm per il giapponese, che utilizza dei caratteri grafici speciali, aggiungendo -a al comando di mc può aiutare a prevenire dei problemi.

4.3.2 Lanciare MC

```
$ mc
```

MC si prende cura di tutte le operazioni sui files tramite il proprio menu, richiedendo il minimo sforzo da parte dell'utente.

4.3.3 File manager

Come default vengono presentate due finestre affiancate che mostrano la lista di files contenuti nelle directory correnti. Un'altra modalità utile è impostare la finestra di destra ad "informazioni", per avere tute le informazioni su file, tipo privilegi di accesso, ecc. A seguire si riportano i tasti fondamentali. Se gpm sta girando, potete usare anche il mouse. (Ricordatevi di premere il tasto shift per avere il normale comportamento sotto MC per taglia ed incolla).

- F1: Menu aiuto
- F3: File viewer interno
- F4: Editor interno
- F9: Attiva il menu a cascata
- F10: Esce da Midnight Commander
- Tab: Muove tra le due finestre

- Insert: Marca il file per operazioni con più files, tipo copia
- Del: Cancella il file (Fate attenzione—impostate MC in modalità cancellazione sicura)
- Tasti cursore: Si spiegano da sè

4.3.4 Trucchi per la riga di comando:

- Qualsiasi comando 'cd' cambierà la directory mostrata sullo schermo selezionato.
- Control-Enter o Alt-Enter copia il nome del file sulla riga di comando. Usatelo insieme ai comandi 'cp' o 'mv'.
- Alt-Tab mostra le scelte per i suffissi di file.
- Si possono stabilire le directory di pertenza per ciascuna finestra, come argomenti per MC; per esempio, "mc /etc /root".
- Esc + tastonumero == Fn (cioè, Esc + 1 = F1, ecc.; Esc + 0 = F10)
- Esc tasto == Alt-key (= Meta, M-); cioè, premete Esc + c per Alt-c

4.3.5 L'editor

L'editor interno ha un sistema di taglia ed incolla interessante. Premendo F3 si marca l'inizio della selezione, un secondo F3 ne segna la fine e la evidenzia. Muovete ora il cursore. Premedo F6 l'area selezionata viene mossa dove è il cursore. Con F5 l'area verrà copiata dove è il cursore. F2 salva il file, F10 esce.

Questo editor può essere lanciato direttamente per un determinato file:

```
$ mc -e file_da_modificare
$ mcedit file_da_modificare
```

Non è un editor multifinestra, si può ottenere lo stesso effetto utilizzando più cosoles. Per copiare da una finestra all'altra, usate la combinazione Alt-Fn-tasto per passare da una console ad un'altra e"File->Insert file" o "File->Copy to file" per muovere parti di un file in un altro.

L'editor interno può essere sostituito da qualsiasi editor esterno preferiate.

Molti programmi usano variabili d'ambiente tipo `EDITOR` o `VISUAL` per decidere quale editor usare. Se vi trovate male con vim, impostatele a "mcedit" aggiungendo queste righe a ~/.bashrc:

```
...
export EDITOR=mcedit
export VISUAL=mcedit
...
```

Raccomando comunque di impostarle a vim, se possibile. Abituarsi ai comandi di vi(m) sarebbe la cosa giusta da fare, dato che sono una costante nel mondo Linux/Unix.

4.3.6 Viewer

E' un viewer evoluto. E' uno strumento prezioso per la ricerca di parole nei documenti. Lo uso sempre per i files nella directory /usr/share/doc. Rappresenta uno dei modi più rapidi di girare tra la massa di informazioni su Linux. il viewer può essere lanciato direttamente:

```
$ mc -v file_da_vedere
```

(Notate che alcuni pacchetti violano le regole e ancora mettono i loro documenti in /usr/doc.)

4.3.7 Auto start

Premete "Enter" su un file, ed il programma appropriato si prenderà cura del suo contenuto. E' una caratteristica di MC molto utile.

```
eseguibile:        Esegue il comando
file man, html:    Dirotta il contenuto al viewer corrispondente
file tar, gz, rpm: Sfoglia il contenuto come fosse una subdirectory
```

Per permettere a queste uilità di svolgere il loro compito, i files da leggere non devono essere segnati come eseguibili. Cambiatene lo stato tramite il coamndo chmod, oppure attraverso il menu file di MC.

4.3.8 File system virtuale FTP

MC può essere utilizzato per accedere a files tramite Internet, usando FTP. Attivate il menu premendo F9, poi attivate il file system virtuale ftp premendo 'p'. Inserite una URL sotto forma di "username:passwd@nomehost.nomedomain", che raggiungerà un directory remota che apparirà come una locale.

4.4 Per studi ulteriori

Molti sono i testi a livello base per Unix. I libri della O'Reilly's in genere sono delle ottime scelte per delle guide in ogni campo riguardante i computers. Raccomando *Unix Power Tools*, ricchissimo di informazioni.

`/usr/share/doc/HOWTO/en-txt/Tips-HOWTO.gz` (LDP) è un'altra risorsa da guardare.

Capitolo 5

Passare a Woody

Le note ufficiali di rilasci per l'aggiornamento si trovano presso http://www.debian.org/releases/stable/releasenotes e http://www.debian.org/releases/testing/releasenotes (lavoro in corso).

5.1 Preparazione

L'aggiornamento a "testing" tramite la rete può essere eseguito come segue (lanciate lo script go-woody (http://qref.sourceforge.net/quick/examples/go-woody) per fare questo con un unico comando):

```
# cd /etc/apt
# cp -f sources.list sources.old
# :>sources.list
# cd /
# apt-setup noprobe
 ... seleziona http o ftp
# cd /etc/apt
# grep -e "^deb " sources.list             >sources.deb
# grep -e "^deb-" sources.list             >sources.src
# sed -e "s/^d/#d/" /usr/share/doc/apt/examples/sources.list >sources.list
# sed -e "s/stable/testing/"  sources.deb >>sources.list
# apt-get update
# apt-get install apt apt-utils
# cat >preferences <<EOF
> Package: *
> Pin: release a=testing
```

```
> Pin-Priority: 700
>
> Package: *
> Pin: release a=unstable
> Pin-Priority: 70
>
> EOF
# sed -e "s/stable/unstable/" sources.deb >>sources.list
# sed -e "s/stable/unstable/" sources.src | \
> sed -e "s/^deb-/#deb-/"                >>sources.list
```

Linee guida per /etc/apt/preferences (vedere “man apt_preferences”):

```
Se sei su stable: cambia la Pin-Priority di testing ad 80
Se sei su testing: mantienilo così com'è (installa unstable da
  /unstable)
Se sei su testing(unstable): cambia la Pin-Priority di unstable a 600
Se sei su unstable(testing): cambia la Pin-Priority di unstable a 800
```

Le linee guida per la scelta della Pin-Priority muovono dall'alto verso il basso nella tabella precedente, man mano che il tempo passa, dal momento immediatamente successivo al rilascio di una distribuzione fino al momento del freeze della successiva.

Impostate apt per l'uso di un proxy impostando la variabile di ambiente `http_proxy`, oppure impostate il file `/etc/apt/apt.conf`.

Il metodo appena descritto aggiorna solamente apt ed un numero limitato di altri pacchetti, per evitare problemi di dipendenze.

Dato che questo metodo di aggiornamento usa `apt-get`, la gestione di *raccomanda* e *suggerisce* è limitata. Leggete ‘Dipendenze’ a pagina 15 ed usate `dselect` per avere un controllo più raffinato.

5.2 Aggiornamento a woody

Dopo aver eseguito la preparazione precedente:

```
# apt-get update # eseguitelo sempre prima dell'upgrade
 ... per aggiornare l'intero sistema seguendo le selezioni suggerite
# apt-get -u dist-upgrade
 ... per aggiornare seguendo le impostazioni correnti di dselect
(nuovo metodo, migliore)
# apt-get -u dselect-upgrade # utilizza le impostazioni di dselect
```

5.3 Configurare Woody

Con i miglioramenti di woody, ciò potrebbe no essere più necessario.

Se avete un sistema woody installato di fresco, modificate `/etc/apt/sources.list`, `/etc /apt/apt.conf`,e `/etc/apt/preferences` per ottenere la stessa struttura descritta nella sezione precedente.

5.4 Ottimizzare sources.list

Ottimizzate sources.list provando ogni sito per latenza e larghezza di banda.

```
# apt-get install apt-spy
# cd /etc/apt ; mv sources.list sources.list.org
# apt-spy -d testing -l sources.apt
```

Capitolo 6

Gestione dei pacchetti in Debian

Abbiate l'accortezza di organizzare un proxy http locale utilizzando "squid", per i pacchetti scaricati da APT. Ciò migliora enormemente la performance degli upgrades via rete, in particolar modo con più macchine Debian su una LAN. Il capitolo si applica sia a sistemi Woody che Potato, tranne che per `apt_preferences(5)`,`/etc/preferences` ed argomenti correlati.

6.1 Introduzione

Se la lettura di tutta la documentazione per gli sviluppatori è troppo per voi, leggete questo capitolo per primo ed iniziate a gustare in pieno il potere della Debian con le distribuzioni testing/unstable :-)

6.1.1 Strumenti fondamentali

```
dselect - strumento di gestione dei pacchetti basato su menu (livello
superiore) apt-get - installa i pacchetti (centrato sull'archivio pacchetti,
APT) dpkg - installa i pacchetti (centrato sui files dei pacchetti) aptitude &
deity - Rimpiazzi futuri di dselect (Interfacce grafiche utente per APT)
```

Questi strumenti non agiscono allo stesso livello. "dselect" gira al di sopra di APT (il comando da terminale è "apt-get) e di "dpkg". Quando usate dselect, accertatevi di aggiornare il file di stato (mediante [U]pdate)prima di selezionare i pacchetti, se li avete installati tramite "apt-get".

Per quanto riguarda le dipendenze, `apt-get` prende automaticamente i pacchetti sotto *dipende*, lasciando quelli sotto *suggerisce* e *raccomanda*, mentre `dselect` offre un controllo più fine sulla scelta dei pacchetti. Vedere 'Dipendenze' a pagina 15.

6.1.2 Strumenti utili

```
apt-cache      - controlla l'archivio dei pacchetti nella cache locale
dpkg-reconfigure  - riconfigura un pacchetto già installato
dpkg-source       - gestisce il pacchetto con il file sorgente
dpkg-buildpackage - automatizza la creazione di un pacchetto
...
```

6.2 Comandi di sopravvivenza Debian

Conoscendoli, potrete vivere una vita di eterni "upgrade" :-)

Fate riferimento anche a 'Installazione di Debian' a pagina 29 e 'Passare a Woody' a pagina 49.

6.2.1 Installare i *task* con "`tasksel`"

"`tasksel`" è definito come l'"*Installatore Debian per i Task*" che viene presentato come una "`semplice`" opzione durante l'installazione del sistema.

Quando si ha la necessità di installare delle funzionalità comuni che richiedono più pacchetti, questo è il modo migliore. Lanciatelo come segue:

```
# dselect update
# tasksel
```

6.2.2 Installare il sistema con APT

Potete installare selettivamente pacchetti provenienti da archivi diversi utilizzando le ultime versioni di apt-get (>woody). Queste permettono di utilizzare file provenienti da "unstable" e "stable", mentre si utilizza "testing".

```
# apt-cache policy libc6 libc6-dev locales
         # controlla lo stato dei pacchetti
# apt-get install libc6=2.2.4-1 libc6-dev=2.2.4-1 locales=2.2.4-1
# apt-get install libc6/unstable libc6-dev/unstable locales/unstable
# apt-get install -t unstable libc6 libc6-dev locales
# apt-get -u install interesting-new-package remove-package-
# apt-get remove useless-old-package
# apt-get remove --purge really-useless-old-package
```

Per riportare tutti i pacchetti ad un livello inferiore (”stable“), è necessario modificare /etc/apt/preferences come segue:

```
Package: *
Pin: release a=stable
Pin-Priority: 1001
```

ed eseguire ”apt-get upgrade“, che forzerà i pacchetti al livello inferiore, in virtù di Pin-priority > 1000.

6.2.3 Aggiornare il sistema con APT

Aggiornamento del sistema con apt:

```
# apt-get update ... poi eseguite uno dei seguenti comandi:
# apt-get -u upgrade # scarica tutti i pacchetti raccomandati
# apt-get -u dist-upgrade # scarica tutti i pacchetti raccomandati
                          # e controlla le dipendenze
# apt-get -u dselect-upgrade # segue la selezione fatta con dselect
```

Utilizzare l’opzione ”-s“ per simulare l’aggiornamento senza realmente effettuarlo.

”dselect“ offre un’interfaccia basata su menù per il controllo di APT. ”deity“ ed ”aptitude“ offriranno delle alternative a ”dselect“.

6.2.4 Controllo dei bachi in Debian

Normalmente gran parte dei problemi sono già stati riconosciuti. Per prima cosa consultate i seguenti siti:

```
$ lynx http://bugs.debian.org/
$ lynx http://bugs.debian.org/<packagename>
```

Cercate su Google (www.google.com, oppure www.google.it) includendo nelle frasi da ricercare ”site:debian.org“.

Se avete ancora dubbi, leggete i manuali (RTFM). Impostate CDPATH come segue:

```
export CDPATH=.:/usr/local:/usr/share/doc
```

e digitate

```
$ cd <packagename>
$ mc <packagename>
```

6.2.5 Aggiornamento con APT: risoluzione dei problemi

Problemi di dipendenze possono accadere quando si esegue l'aggiornamento in unstable/testing. Molte volte ciò accade poichè un pacchetto da aggiornare presenta delle nuove dipendenze che non vengono soddisfatte. Problemi del genere vengono risolti usando

```
# apt-get dist-upgrade
```

Se questo non funziona, allora ripetete uno dei seguenti comandi finchè il problema non si risolve da sè:

```
# apt-get upgrade -f      # continua l'aggiornamento nonostante l'errore
... oppure
# apt-get dist-upgrade -f # continua il dist-upgrade nonostante l'errore
```

Alcuni script di aggiornamento realmente difettosi possono causare problemi ricorrenti. Per risolvere tali situazioni è in genere buona cosa ispezionare gli script /var/lib/dpkg/info/packagename.{post-,pre-}{install,removal} del pacchetto responsabile e poi eseguire:

```
# dpkg --configure -a
        # configura tutti i pacchetti installati parzialmente
```

Se uno script lamenta la mancanza di un file di configurazione, date un'occhiata in /etc per il file di configurazione corrispondente. Se ne esiste uno con l'estensione .new (o qualcosa di simile), cambiatelo (mv) per rimuovere il suffisso.

Problemi di dipendenze possono accadere installando i pacchetti in unstable/testing. Esistono dei metodi per aggirare le dipendenze.

```
# apt-get install -f package # ignora le dipendenze difettose
```

Un metodo alternativo consiste nell'utilizzare il pacchetto `equivs`. Vedere 'Pacchetto `equivs`' a pagina 63.

6.2.6 Salvataggio mediante dpkg

Un recupero ad hoc di un dselect (APT) che non funziona più può essere effettuato su un sistema pesantemente malfunzionante, semplicemente usando dpkg senza APT:

```
# cd /var/cache/apt/archives
# dpkg -i libc6* libdb2* perl*
# dpkg -i apt* dpkg* debconf*
# dpkg -i *  (finchè non compaiono più errori)
```

Se manca un pacchetto, recuperatelo con:

```
# mc    # usate "FTP link" all'indirizzo di un server Debian FTP
```

Fino a questo momento, i pacchetti sul server http/ftp potrebbero non essere contenuti nella directory classica /dist, ma sotto la nuova directory /pool.

Poi installatelo con:

```
# dpkg -i /var/cache/apt/archives/packagefile.deb
```

Se vi sono dipendenze non risolte, risolvetele oppure usate:

```
# dpkg --ignore-depends=package1,... -i packagefile.deb
# dpkg --force-depends -i packagefile.deb
# dpkg --force-depends --purge package
```

6.2.7 Installare un pacchetto su un sistema non bootabile

Fate il boot in Linux mediante un floppy o CD di salvataggio Debian, o tramite una partizione alternativa se avete un sistema con più boot in Linux. Montate il sistema non bootabile su /target ed usate il modo chroot di installazione di dpkg.

```
# dpkg --root /target -i packagefile.deb
```

Poi configurate il tutto e risolvete i problemi.

A proposito, se tutto quello che impedisce il boot è un lilo malfunzionante, potete fare il boot con un disco di salvataggio Debian standard. Al prompt del boot, ammettendo che la partizione root della vostra installazione è su /dev/hda12 e volete un runlevel 3, digitate:

```
boot: rescue root=/dev/hda12 3
```

A questo punto avrete accesso ad un sistema pressochè perfettamente funzionante con il kernel su dischetto. (Potranno esserci inconvenienti minori dovuti alla mancanza di moduli o caratteristiche del kernel.)

6.3 Comandi del "paradiso" Debian

6.3.1 Informazioni sui file

Per scoprire a quale pacchetto un file appartiene:

```
$ dpkg {-S|--search} file (o modello)
        # trova il pacchetto a partire dal nome del file installato
$ zgrep -e file (o modello) /local/copy/of/debian/woody/Contents-i386.gz
        # trova l'elenco dei file con la stessa radice (o il file)
 # nell'archivio Debian
```

Potete utilizzare anche dei comandi specifici per i pacchetti:

```
# apt-get install dlocate # è in conflitto con slocate
                          # (la versione sicura di locate)
$ dlocate filename # alternativa rapida a dpkg -L e dpkg -S
...
# apt-get install auto-apt
        # strumento per l'installazione a richiesta dei pacchetti
# auto-apt update # crea un database per auto-apt
$ auto-apt search nomefile (o modello)
        # effettua una ricerca completa nell'archivio pacchetti
...
```

6.3.2 Informazioni sui pacchetti

Cerca le informazioni negli archivi dei pacchetti e le visualizza. Accertatevi che apt sia indirizzato verso l'archivio(i) appropriato(i) modificando `/etc/apt/sources.list`. Se volete vedere le differenze fra i pacchetti in testing/unstable e quelli correntemente installati utilizzate "apt-cache policy"—, un comando molto utile.

```
# apt-get check # aggiorna la cache e controlla le dipendenze
$ apt-cache search testo # cerca un pacchetto a partire dalla descrizione
$ apt-cache policy pacchetto # fornisce informazioni su priorità e
       # distribuzione di appartenenza di un pacchetto
$ apt-cache show -a pacchetto # mostra la descrizione di un pacchetto
                              # per ogni distribuzione
$ apt-cache showpkg pacchetto # Informazioni per il debug di un
                              # pacchetto
# dpkg --audit|-C          # cerca i pacchetti parzialmente installati
$ dpkg {-s|--status} pacchetto... # stato e descrizione del pacchetto
                              # installato
$ dpkg -l pacchetto ... # stato del pacchetto installato (1 linea)
$ dpkg -L pacchetto ... # elenca i file installati per un dato pacchetto
```

E' sempre possibile trovare informazioni su pacchetti in (Io uso "mc"):

```
/var/lib/apt/lists/*
/var/lib/dpkg/{available|status}
```

6.3.3 Riconfigurare i pacchetti già installati

Usate i seguenti comandi per riconfigurare un pacchetto già installato.

```
# dpkg-reconfigure --priority=medium pacchetto [...]
# dpkg-reconfigure --all  # riconfigura tutti i pacchetti
```

Eseguite questo tramite debconf se avete la necessità di cambiarne la configurazione dei messaggi in modo permanente.

Alcuni programmi hanno degli script speciali di configurazione.

```
apt-setup     - crea /etc/sources.list
install-mbr   - installa un manager di Master Boot Record
tzconfig      - imposta il fuso orario locale
gpmconfig     - imposta il demone per mouse gpm
smbconfig     - configura Samba
eximconfig    - configura Exim (MTA)
texconfig     - configura teTeX
apacheconfig  - configura Apache (httpd)
cvsconfig     - configura CVS
```

```
sndconfig     - configura il sistema audio
...
update-alternatives - imposta i comandi predefiniti, p.e. vim per vi
update-rc.d         - Gstione degli script di inzializzazione del System-V
update-menus        - Il sistema dei menu Debian
...
```

6.3.4 Rimozione e purga dei pacchetti

Rimuove un pacchetto mantenendone la configurazione:

```
# apt-get remove pacchetto ...
# dpkg  --remove pacchetto ...
```

Rimuove un pacchetto e la sua configurazione:

```
# apt-get remove --purge pacchetto ...
# dpkg    --purge        pacchetto ...
```

6.3.5 Mantenere vecchi pacchetti

Per esempio, per mantenere libc6 e libc6-dev con ”dselect“ e ”apt-get -u upgrade package“ si può procedere come segue:

```
# echo -e "libc6 hold\nlibc6-dev hold" | dpkg --set-selections
```

”apt-get -u installa pacchetto“ non verrà impedito da quell”’hold“. Per mantenere un pacchetto forzandone un declassamento automatico con ”apt-get -u upgrade pacchetto“oppure ”apt-get -u dist-upgrade“, aggiungete le seguenti linee ad /etc/apt/preferences:

```
Package: libc6*
Pin: release a=stable
Pin-Priority: 2000
```

Il comando seguente mostra i pacchetti mantenuti (non aggiornati):

```
dpkg --get-selections "*"|grep -e "hold$"
```

6.3.6 Dselect - configurazione generale

Aggiungete una linea con "expert" in /etc/dpkg/dselect.cfg per ridurne il "rumore".

Una volta lanciato, dselect seleziona automaticamente tutti i pacchetti "Required", "Important", e "Standard". Alcuni pacchetti come teTEX ed emacs, sono voluminosi e potrebbero non essere necessari durante la prima installazione di un sistema, per cui siate accorti.

Dselect presenta un'interfaccia utente un pò strana. Quattro sono i comandi ambigui (Maiuscolo significa MAIUSCOLO!):

```
Tasto  Azione
Q      Esci. Conferma la selezione corrente ed esce comunque
       (prevale su le dipendenze)
R      Torna indietro. Non intendevo farlo.
D      Dannazione! Non mi interessa cosa ne pensa dselect.
       Fallo e basta!
U      Riporta tutto allo stato sUggerito
```

Usando D e Q, potete scegliere delle selezioni che presentano conflitti di dipendenze a vostro rischio e pericolo. Usate questi comandi con cura. Per macchine lente, lanciate "dselect" su un'altra macchina più veloce per trovare i pacchetti che vi servono ed usate "apt-get install" per installarli. "apt-get dselect-upgrade" è il comando che utilizza al meglio la selezione con dselect.

6.3.7 Ridurre la dimensione dei files nella cache

L'installazione dei pacchetti tramite apt lascia i loro files archiviati in `/var/cache/apt/archives` e questi vanno rimossi.

```
# apt-get autoclean # rimuove solo pacchetti inutili
# apt-get clean     # rimuove tutti i pacchetti nella cache
```

6.3.8 Salvare/copiare la configurazione del sistema

Per fare una copia locale dello stato della selezione dei pacchetti:

```
$ dpkg --get-selections "*" >lemieselezioni   # oppure usare \*
```

"*" fa includere anche i nomi dei pacchetti marcati con "purge".

Potete poi trasferire questo file su un altro computer ed installarlo con:

```
# apt-get update
# dpkg --set-selections <lemieselezioni
# apt-get -u dselect-upgrade
```

6.3.9 Portare un pacchetto nel sistema "stable"

Se si eseguono degli aggiornamenti parziali di un sistema "stable", può essere buona cosa ricreare un pacchetto all'interno del proprio ambiente a partire dai sorgenti. In tal modo si eviteranno dei massicci aggiornamenti di pacchetti, dovuti alle loro dipendenze. Per prima cosa, aggiungete le linee seguenti a /etc/apt/sources.list:

```
deb-src http://http.us.debian.org/debian testing main contrib non-free
deb-src http://non-us.debian.org/debian-non-US testing/non-US main
contrib non-free
deb-src http://http.us.debian.org/debian unstable main contrib non-free
deb-src http://non-us.debian.org/debian-non-US unstable/non-US main
contrib non-free
```

Poi recuperate i sorgenti e costruite un pacchetto locale:

```
$ apt-get source package/unstable
$ dpkg-source -x package.dsc
$ cd package-version
  ... controlla i pacchetti necessari (Build-depends nel file
.dsc) e li installa. Avete bisogno anche del pacchetto fakeroot.

$ dpkg-buildpackage -rfakeroot

  ...oppure (senza "signature")
$ dpkg-buildpackage -rfakeroot -us -uc
# utilizzate "debsign" in seguito, se necessario

  ...oppure (senza "signature")
# ./debian/rules binary
# ./debian/rules clean
# cd ..
# dpkg-source -b package-version

  ...poi per installare il pacchetto
# dpkg -i packagefile.deb
```

Normalmente, per soddisfare le dipendenze si ha la necessità di installare alcuni pacchetti con il suffisso "-dev". "debsign" è contenuto nel pacchetto "devscripts". "auto-apt" può aiutare nel soddisfare queste dipendenze.

In woody, questi problemi di dipendenze possono essere semplificati. Per esempio per compilare solo il sorgente del pacchetto `pine`:

```
# apt-get build-dep pine
# apt-get source -b pine
```

6.3.10 Convertire od installare un pacchetto binario di altra distribuzione

`alien` permette la conversione dei pacchetti binari nei formati Redhat rpm, Stampede slp, Slackware tgz, e Solaris pkg, in pacchetti Debian deb. Se volete usare un pacchetto proveniente da un'altra distribuzione di Linux al posto di auello installato sul vostro sistema, potete utilizzare alien per convertirlo nel formato che più preferire. Alien supporta anche pacchetti LSB.

6.4 Altre particolarità di Debian

6.4.1 dpkg-divert

`dpkg-divert` forza `dpkg` a non installare un file nella sua localizzazione normale, ma in una posizione *diversa*.

Gli amministratori di sistema od anche script di manutenzione, possono usare questo comando per scavalcare alcuni files di configurazione dei pacchetti, oppure quando è necessario che dei files (non marcati come `conffiles`) devono comunque essere conservati da `dpkg` quando si installa una nuova versione del pacchetto che contiene quei files da conservare.

```
# dpkg-divert [--add]  nomefile # aggiunge "divert"
# dpkg-divert --remove nomefile # rimuove "divert"
```

6.4.2 Pacchetto `equivs`

Se compilate un programma dai sorgenti, è meglio debianizzarlo in un pacchetto (*.deb). Usate `equivs` solo come ultima risorsa.

```
Package: equivs
Priority: extra
```

```
Section: admin
Description: Aggira le dipendenze dei pacchetti Debian E' un pacchetto
 vuoto che può essere usato per creare dei pacchetti Debian contenenti
 solo le informazioni sulle dipendenze.
```

6.4.3 Comandi alternativi

Per lanciare "vim" con il comando "vi", usate "update-alternatives":

```
# update-alternatives --display vi
...
# update-alternatives --config vi
  Selection    Command
-----------------------------------------------
      1        /usr/bin/elvis-tiny
      2        /usr/bin/vim
*+    3        /usr/bin/nvi

Enter to keep the default[*], or type selection number: 2
```

Le alternative del sistema in Debian sono contenute in /etc/alternatives come link simbolici.

Per impostare il vostro window manager preferito, usate invece "x-window-manager".

"/bin/sh" è un link simbolico a "/bin/bash" o "/bin/ash/". E' più sicuro usare "/bin/bash" per mantenere la compatibilità con vecchi script in bash, ma più corretto utilizzare "/bin/ash" per imporre la compatibilità POSIX.

6.4.4 Inizializzazione e livelli (runlevels) del System-V

Il runlevel predefinito in cui il sistema esegue il boot può essere impostato in /etc/inittab.

Diversamente da altre distribuzioni, Debian pone la gestione dei livelli completamente nelle mani dell'amministratore di sistema. La gestione dell'inizializzazione del System-V in Debian viene eseguita tramite gli scripts update-rc.d.

Lanciare /etc/init.d/nomescript nel runlevel 1,2 o 3 e terminarlo nel 4 o 5 con una priorità di sequenza 20 (normale) può essere fatto mediante:

```
# update-rc.d nomescript start 20 1 2 3 . stop 20 4 5 .
```

I link simbolici possono essere rimossi mentre lo script è ancora presente in init.d tramite:

```
# update-rc.d -f nomescript remove
```

Per modificare i runlevels, io baro. Ne cambio le voci manualmente usando il comando "mv" allo shell prompt di "mc", copiandone i link con "Alt-Enter". Sono giunto persino a disabilitare un demone inserendo "exit 0" all'inizio di uno script init.d. Dopo tutto, questi sono dei semplici file di configurazione.

6.4.5 Demoni di servizio disabilitati

La distribuzione Debian ha molto a cuore la sicurezza del sistema e si aspetta che l'amministratore sia uno realmente competente al riguardo. Per questo motivo, talvolta la facilità d'uso diventa un problema secondario rispetto alla sicurezza e molti demoni di servizio sono impostati al massimo della sicurezza - con il minimo, o nessun servizio disponibile di default.

Se avete dei dubbi (exim, dhcp, ...) controllate `ps aux` oppure il contenuto di `/etc/init.d/*` e di `/etc/inetd.conf`. Controllate anche `/etc/hosts.deny` in 'Controllo del login' a pagina 79.

X11 non permette connessioni TCP/IP (remote) di default nelle versioni più recenti di Debian. Vedere 'Connessione TCP/IP ad X' a pagina 93.

Capitolo 7

Il kernel Linux su Debian

Debian ha i suoi modi per la ricompilazione del kernel e dei relativi moduli.

7.1 Ricompilare il kernel

7.1.1 Il metodo standard Debian

Usate il nuovo "kernel-package" in unstable.(7/2001) Anche "tar" ha subito alcuni cambiamenti nel passaggio da potato a woody, per cui utilizzate -j al posto di -I per i files .bz2. Controllate anche i bug reports per gcc, binutils e modutils.

La compilazione di un proprio kernel dai sorgenti in un sistema Debian richiede una cura speciale. Per compilare più "kernel-images", utilizzate il nuovo "–append_to_version" con make-kpkg.

```
# apt-get install debhelper modutils kernel-source-2.4.12 kernel-package
# vi /etc/kernel-pkg.conf    # inserisco il mio nome ed email
$ cd /usr/src                # la directory di compilazione
... scaricate da kernel.org, oppure installate i pacchetti
... kernel-*-source.deb.
$ tar -xvjf kernel-source-2.4.12.tar.bz2
$ cd kernel-source-2.4.12    # se è il vostro sorgente
$ rm -rf */pcmcia
       # [FACOLTATIVO] se volete usare i moduli da pcmcia-cs
$ cp /boot/config-2.4.12-386 .config
       # usa il config corrente come default
$ make menuconfig            # customizzate a vostro piacimento
$ make-kpkg clean            # deve essere eseguito (per: man make-kpkg)
```

```
$ fakeroot make-kpkg --append_to_version -486 --initrd \
        --revision=rev.01 kernel_image
$ cd ../modules/pcmcia
$ fakeroot ./debian/rules
$ cd ../..
# dpkg -i kernel-image*.deb pcmcia-cs*.deb # install
```

"make-kpkg kernel_image" esegue in raltà un "make oldconfig" ed un "make dep" Se initrd non è usato, non mettete –initrd.

Si può evitare di dover dare "rm -fr */pcmcia" selezionando "General setup —>" e "PCMCIA/CardBus support —>" in "make menuconfig" ed impostando la configurazione come "< > PCMCIA/CardBus support".

7.1.2 Il metodo classico

Scaricate i sorgenti originali da:

```
linux:     http://www.kernel.org/
pcmcia-cs: http://pcmcia-cs.sourceforge.net/
```

oppure utilizzate i sorgenti equivalenti da debian ed eseguite i seguenti comandi:

```
# cd /usr/src
# tar xfvz linux-xxx.tar.gz
# rm -rf linux
# ln -s linux-xxx linux
# tar xfvz pcmcia-cs-xxx.tar.gz
# ln -s pcmcia-cs-xxx pcmcia
# cd linux
# rm -rf */pcmcia
        # [FACOLTATIVO] se volete usare i moduli da pcmcia-cs
# make menuconfig
... customizzate a vostro piacimento ...
# make dep
# make bzImage
... modifiche per lilo / grub ...
... muovete /usr/src/linux/arch/i386/boot/bzImage in boot ...
... /sbin/lilo o qualunque altra cosa serva per grub
# make modules; make modules_install
# cd ../pcmcia
```

```
# make config
# make all
# make install
... aggiungete i nomi dei moduli richiesti a /etc/modules
# shutdown -r now
... fate il boot con il nuovo kernel ...
```

L'uso di gcc, binutils, modutils prese da Debian unstable può essere di aiuto.

7.2 Kernel 2.4 modulare

I nuovi kernel debian 2.4 sono altamente modularizzati. Per farli funzionare come avete deciso, dovrete essere sicuri che i moduli siano veramente lì.

7.2.1 PCMCIA

/etc/modules deve contenere i seguenti driver, affinchè PCMCIA funzioni:

```
# Driver ISA PnP
isa-pnp
# Driver PCMCIA di basso livello
# yenta_socket # non sembra essere necessario :-(
```

Il resto viene configurato dagli scripts pcmcia (del pacchetto pcmcia-cs) da depmod e kmod. Credo che isa-pnp mi sia stato necessario perchè il mio laptop è un vecchio isa-pcmcia. I modelli più recenti con cardbus-pcmcia non lo richiedono.

[Da Miquel van Smoorenburg <miquels@cistron.nl>]

"Ho semplicemente rimosso tutta la roba correlata con pcmcia dal mio laptop, compreso cardmgr, ecc., ed ho installato un kernel 2.4 con supporto per cardbus, insieme al nuovo pacchetto "hotplug" da woody.

Se avete delle PC card a 32 bit non avete bisogno del pacchetto pcmcia, il kernel 2.4 ha i servizi per le card già inseriti. Il driver tulip standard dovrebbe funzionare bene con la vostra card dlink.

—Mike."

7.2.2 SCSI

[NON TESTATO]/etc/modules deve contenere i seguenti driver affinchè SCSI funzioni:

```
# SCSI core
scsi_mod
# Driver generico SCSI
sg
# Disco SCSI
sd_mod
# Tutti gli altri moduli hardware necessari
...
```

Forse depmod può prendersi cura di alcuni dei moduli summenzionati.

7.2.3 Funzioni di rete

/etc/modules deve contenere i seguenti driver per le funzioni extra di rete:

```
# net/ipv-4
ip_gre
ipip

# net/ipv-4/netfilter
# iptable (nell'ordine)
ip_tables
ip_conntrack
ip_conntrack_ftp
iptable_nat
iptable_filter
iptable_mangle
#
ip_nat_ftp
ip_queue
#
ipt_LOG
ipt_MARK
ipt_MASQUERADE
ipt_MIRROR
ipt_REDIRECT
```

```
ipt_REJECT
ipt_TCPMSS
ipt_TOS
ipt_limit
ipt_mac
ipt_mark
ipt_multiport
ipt_owner
ipt_state
ipt_tcpmss
ipt_tos
ipt_unclean
#
#ipchains
#ipfwadm
```

Quanto sopra non è ottimizzato al meglio. Depmod potrebbe prendersi cura di alcuni dei moduli.

7.2.4 Il filesystem EXT3 (> 2.4.17)

L'attivazione di un filesystem "journaled" con il tipo EXT3 FS richiede i seguenti passi, usando una kernel-image Debian precompilata (> 2.4.17):

```
# cd /etc; mv fstab fstab.old
# sed 's/ext2/ext3,ext2/g' <fstab.old >fstab
# vi /etc/fstab
... imposta il filesystem di root ad "auto" invece di "ext3,ext2"
# cd /etc/mkinitrd
# echo jbd >>modules
# echo ext3 >>modules
# echo ext2 >>modules
# cd /
# apt-get update; apt-get install kernel-image-2.4.17-686-smp
... installa l'ultimo kernel ed imposta il boot
    (lilo viene lanciato a questo punto)
# tune2fs -j -i 0 /dev/hda1
# tune2fs -j -i 0 /dev/hda2
... Per tutti i filesystems EXT2 convertiti a EXT3
# shutdown -r now
```

A questo punto il EXT3 journaling è attivato. Avere le voci in `fstab` come `ext3,ext2` assicura, in caso il kernel non supportasse ext3, un tranquillo ritorno ad ext2 per le partizioni non-root.

Se avete un kernel 2.4 già installato e non lo volete reinstallare, date gli stessi comandi `apt-get` di cui sopra e poi:

```
# mkinitrd -o /boot/initrd.img-2.4.17-686-smp /lib/modules/2.4.17-686-smp
# lilo
# tune2fs -j -i 0 /dev/hda1
# tune2fs -j -i 0 /dev/hda2
... per tutti i filesystems EXT2 convertiti a EXT3
# shutdown -r now
```

A questo punto il journaling EXT3 è attivato.

Qualora `/etc/mkinitrd/modules` non fosse stato impostato al lancio di `mkinitrd` e si volessero aggiungere alcuni moduli al boot:

```
... al prompt di initrd per entrare in shell (entro 5 sec.), premete RETURN
# insmod jbd
# insmod ext3 # modprobe ext3 si prende cura di tutto
# insmod ext2
# ^D
... continuate il booting
```

Sui messaggi al boot (dmesg), apparirà "cramfs: wrong magic", messaggio noto per essere inoffensivo. (Verrà risolto nel rilascio successivo).

Alcuni sistemi sperimentano dei gravi blocchi del kernel, quando EXT3 è attivato (2.4.17).

7.2.5 Supporto nel kernel 2.4 per Realtek RTL-8139

Per qualche motivo, il modulo per RTL-8139 non si chiama più rtl8139, ma 8139too. All'aggiornamento del kernel da 2.2 a 2.4, dovrete modificare a mano `/etc/modules` per inserire il nuovo nome del modulo.

Capitolo 8

Trucchi per Debian

8.1 Il sistema di boot

Leggere `/usr/share/doc/HOWTO/en-txt/BootPrompt-HOWTO.txt.gz` (LDP) per informazioni dettagliate sul prompt del boot.

8.1.1 Ho scordato la password di root (1)

Il sistema può comunque essere lanciato nell'account di root, anche se non se ne conosce la password, basta avere accesso alla tastiera. (Ciò significa che non esistono altre password richieste dal BIOS, o dal boot-loader (tipo lilo) che possono prevenire l'accesso al sistema).

Questa procedura non richiede alcun disco di boot, nè cambiamenti al BIOS. In questo contesto, "Linux" è l'etichetta per lanciare il kernel in una installazione Debian standard.

Alla schermata di boot di lilo (in alcuni sistemi si deve premere il tasto shift), date:

```
boot: Linux init=/bin/sh
```

Il sistema lancia il kernel ed esegue /bin/sh al posto dello standard init. A questo punto avete ottenuto i privilegi di root e la shell di root. Siccome, però, / è montata in sola lettura e molte altre partizioni non sono state ancora montate, avete bisogno di eseguire i passi seguenti per avere un sistema ragionevolmente funzionante.

```
init-2.03# mount -n -o remount,rw /
init-2.03# mount -avt nonfs,noproc,nosmbfs
init-2.03# cd /etc
init-2.03# vi passwd
init-2.03# vi shadow
```

Una voce vuota nella seconda colonna di uno di questi files, disabilita la password. Diversamente da alcuni vecchi Linux, la debian potato richiede la password solo se si esegue il boot nel runlevel 1.

8.1.2 Ho scordato la password di root (2)

Fate il boot da qualunque set di dischi di emergenza boot/root. Se, per esempio, /dev/hda3 è la partizione di root originale, i seguenti comandi permetteranno di aprire il file password facilmente come sopra.

```
# mkdir fixit
# mount /dev/hda3 fixit
# cd fixit/etc
# vi shadow
# vi passwd
```

Il vantaggio di questo approccio rispetto al metodo precedente è che non richiede la conoscenza della password di lilo. Però, bisogna essere in grado di settare il boot del sistema da floppy o da CD.

8.1.3 Non riesco a fare il boot

Nessun problema. Nessuno si prende la briga di fare un dischetto di boot durante l'installazione. Se lilo non funziona, usate come disco di boot il disco di installazione di Debian (il primo). eseguite il boot da questo disco. Al prompt, data la partizione di root in /dev/hda12, e runlevel 3 quello che volete, date:

```
boot: rescue root=/dev/hda12 3
```

A questo punto siete entrati in un sistema pressochè funzionante in toto, basato sul kernel del dischetto. (Potrete avere piccole noie, dovute alla mancanza di alcuni moduli).

Se volete crearvi un vostro boot floppy, leggete il readme.txt sul disco di rescue.

8.1.4 Altri trucchi per il boot prompt

Potete fare il boot in un particolare runlevel e configurazione tramite il prompt di lilo. I dettagli li potete trovare nel BootPrompt-HOWTO.txt.gz.

Se volete fare il boot nel runlevel 4, date il seguente comando al prompt di lilo.

```
boot: Linux 4
```

Se si vuole il boot in modalità singolo utente, conoscendo la password di root, potete usare uno dei seguenti esempi.

```
boot: Linux S
boot: Linux 1
boot: Linux -s
```

Se si vuole il boot con meno memoria di quanta il sistema ne abbia (mettiamo 48M in un sistema con 64M), basta dare il seguente comando al prompt.

```
boot: Linux mem=48M
```

Evitate di specificare più memoria di quanta ne abbiate, altrimenti il kernel si pianterà inevitabilmente. Se si hanno più di 64M, per esempio 128M, con vecchi kernel o BIOS bisogna specificare il comando mem=128M al prompt, altrimenti non verranno usati più di 64M.

8.2 Registrazione degli eventi

8.2.1 Registrare gli eventi della shell

L'amministrazione di sistema in ambiente Unix è rappresentata da una serie di compiti molto più elaborati che in un ambiente di un normale PC. E' importante conoscere i modi di configurazione basilari, in caso di soluzione dei problemi del sistema. Le interfacce grafiche di configurazione basate su X sono carine e convenienti, ma comunque limitate, specialmente in queste occasioni.

In questo contesto, la registrazione degli eventi della shell è una buona pratica, specialmente come root.

Emacs: Usate "M-x shell" per iniziare la registrazione in un buffer e "C-x C-w" per scrivere il buffer su un file.

Shell: Usate il comando script.

```
$ script
Script started, file is typescript
 .... fate quello che dovete ...
 Control-D
$ col -bx <typescript >savefile
$ vi savefile
```

Si può usare il seguente invece di script.

```
$ bash -i 2>&1 | tee typescript
```

8.2.2 Registrare gli eventi di X

Se dovete registrare l'immagine di una applicazione di X, incluso x-term, usate gimp. Può catturare ogni singola finestra, oppure lo schermo intero. Un'alternativa è xwd (da console).

8.3 Copia ed archiviazione di una intera subdirectory

8.3.1 Comandi base per copiare una intera subdirectory

Se dovete riarrangiare l'organizzazione delle directory, muovete il contenuto, links compresi con:

```
Metodo standard:
# cp -a /partenza/directory /destinazione/directory # deve essere GNU
# (cd /partenza/directory && tar cf - . ) | \
        (cd /destinazione/directory && tar xvfp - )
Se sono presenti delle hardlinks, è necessario un metodo più "pedante":
# cd /percorso/alla/vecchia/directory
# find . -depth -print0 | afio -p -xv -0a /mount/point/della/nuova/directory
Da remoto:
# (cd /partenza/directory && tar cf - . ) | \
        ssh user@host.dom (cd /destinazione/directory && tar xvfp - )
Se non ci sono links:
# scp -pr user1@host1.dom:/partenza/directory \
          user2@host2.dom:/destinazione/directory
```

Qui, `scp` **<==>** `rcp` e `ssh` **<==>** `rsh`.

Il metodo per copiare una intera subdirectory è basato sulle informazioni fornite da Manoj Srivastava <srivasta@debian.org> sulla lista debian-user@lists.debian.org.

8.3.2 cp

Tradizionalmente, cp non è mai stato un gran candidato per tali scopi, poichè non dereferenzia i link simbolici, nè preserva gli hard links. Un'altro fattore da considerare è lo spezzettamento dei files.

GNU cp ha superato questi limiti; comunque su un sistema non GNU, cp potrebbe avere ancora problemi. In più, usando cp non si possono creare archivi che siano piccoli e facilmente trasportabili..

```
% cp -a . newdir
```

8.3.3 tar

Tar è andato oltre alcuni dei problemi che aveva cp con i links simbolici. Comunque sia, 'cpio' è in grado di trattare files speciali, mentre il tradizionale 'tar' no.

Il modo in cui 'tar' tratta hard links multi ad un file è di porre solo una copia del link sul nastro, ma il nome colegato a quella copia è il *solo* utilizzabile per recuperare il file; 'cpio' invece, mette una copia per ciascun link, ma per recuperare il file potete usare uno qualsiasi dei nomi.

8.3.4 pax

E' la nuova utilità POSIX (IEEE Std 1003.2-1992, pagine 380-388 (sezione 4.48) e pagine 936-940 (sezione E.4.48)), per lo scambio di archivi portabili, nuova di zecca. Pax legge, scrive ed elenca i membri di un file archivio, e ne copia le directories secondo le gerarchie. Le operazioni di pax sono indipendenti dal formato dell'archivio e supportano una grossa varietà di formati diversi.

le implementazioni di pax sono ancora molto nuove (si può dire che puzzano ancora di latte...).

```
$ pax -rw -p e . newdir oppure $ find . -depth | pax -rw -p e newdir
```

8.3.5 cpio

copia i files in un archivio cpio o tar. L'archivio può essere un'altro file sul disco rigido, un nastro magnetico, oppure una pipe.

```
$ find . -depth -print0 | cpio --null --sparse -pvd new-dir
```

8.3.6 afio

Afio è un metodo migliore per trattare con archivi in formato cpio. In gener è più veloce di cpio, ha maggiori opzioni per i nastri e tratta in maniera più gentile con le corruzioni dei dati in entrata. Supporta gli archivi multi-volume durante le operazioni in modalità interattiva. Afio può creare archivi molto più sicuri di quelli compressi con tar o cpio. Afio è anche ottimo come 'motore per l'archiviazione' negli script di backup.

```
$ find . -depth -print0 | afio -px -0a new-dir
```

Tutti i miei backup su nastro usano afio.

8.4 Modificare i files con le espressioni regolari

Sostitusce tutte le istanze di FROM_REGEX con TO_REGEX in tutti i files FILES . . . :

```
# perl -i -p -e 's/FROM_REGEX/TO_REGEX/g;' FILES ...
```

"-i" sta per "editing sul-posto", "-p" sta per "loop implicito tra i FILES . . . ".

8.5 Recuperare un sistema bloccato

8.5.1 Uccidere un processo

Lanciate "top" per vedere quale processo si comporta in maniera "sconveniente". Premete "P" per ordinarli per tempo di cpu, "M" per consumo di memoria e "k" per uccidere il processo.

Usate "kill" e "killall" per uccidere (o mandare il segnale) al processo, per nome.

```
 1: HUP,  riavvia il demone
15: TERM, uccisione normale
 9: KILL, uccisione "dura".
```

8.5.2 ALT-SysRQ

L'assicurazione contro i malfunzionamenti di sistema è un'opzione di compilazione del kernel, denominata "Magic SysRQ key". Premendo ALT-SysRQ su un sistema i386, mediante una sola chiave 'r0keisub', si ottiene la "magia".

un'R'aw riavvia la tastiera quando cose tipo X si bloccano. Cambiare il loglevel a 0 riduce i messagi di errore. sa'K' (system attention key) uccide tutti i processi sulla console virtuale attuale. t'E'rm uccide tutti i processi su terminale corrente tranne init. k'I'll uccide tutti i processi, tranne init.

'S'ync, 'U'mount, e re'B'oot sono utili per uscire da situazioni veramente spinose.

I kernel debian standard non sono compilati al momento con questa opzione attivata. Ricompilate il kernel per farlo. Informazioni dettagliate le trovate in `/usr/share/doc/kernel-doc-version /Documentation/sysrq.txt.gz`.

8.6 Files di configurazione

8.6.1 Controllo del login

```
/etc/securetty           # controlla il root login da console
/etc/hosts.{allow|deny}  # controlla il login dall'host
/etc/pam.d/login         # controlla il login
```

Decommentate "ALL: PARANOID" in /etc/hosts.deny.

Cambiate il contenuto di /etc/pam.d/login come segue, se desiderate dei terminal senza password, ma assolutamente insicuri. A vostro rischio e pericolo.

```
#auth       required   pam_unix.so nullok
auth       required   pam_permit.so
```

nfs ecc. richiede il login per sunrpc. "man 8 portmap".

8.6.2 Defaults

```
/etc/defaults/rcS        # defaults di boot (motd, sulogin,...)
/etc/syslog.conf         # modalità di log del sistema
                           (verificate il pacchetto "colorize")
```

8.7 Alcuni piccoli comandi da ricordare

8.7.1 Memoria libera

"free" e "top" danno buone informazioni sulle risorse di memoria. Non vi preoccupate sulle dimensioni della memoria "used" nella riga "Mem:", ma leggete quella sotto. (38792 nell'esempio qui sotto).

```
$ free -k # for 256MB machine
             total       used       free     shared    buffers cached
Mem:        257136     230456      26680      45736     116136 75528
-/+ buffers/cache:      38792     218344
Swap:       264996          0     264996
```

La memoria fisica esatta può essere confermata da "grep '^Memory' /var/log/dmesg" che è "Memory: 256984k/262144k available (1652k kernel code, 412k reserved, 2944k data, 152k init)".

```
Total         = 262144k = 256M (1k=1024, 1M=1024k)
Free at dmesg = 256984k = Total - kernel - reserved - data - init
Free at shell = 257136k = Total - kernel - reserved - data
```

Circa 5MB non sono utilizzabili dal sistema, perchè a disposizione del kernel.

8.7.2 Impostare l'ora (BIOS)

```
# date MMDDhhmmCCYY
# hwclock --utc
# hwclock --systohc
# hwclock --show
```

Questo imposterà l'orologio del sistema e dell'hardware a MM/DD hh:mm, CCYY. Le ore sono allineate al fuso locale, ma l'orologio dell'hardware utilizza lo "Universal Time" (UT).

Coorezione automatica dell'ora di sistema da un server remoto:

```
# ntpdate <server>
```

Se avete una connessione permanente, è bene averlo in `/etc/cron.daily`.

8.7.3 Come disabilitare lo screensaver

In console:

```
# setterm -powersave off
```

Lanciate kon2(kanji)con:

```
# kon -SaveTime 0
```

In X:

```
# xset s off
 o
# xset -dpms
 o
# xscreensaver-command -prefs
```

Leggete le pagine man corrispondenti.

8.7.4 Disabilitare il suono (beep)

Potete sempre staccare gli altoparlanti del PC...;-) Da shell:

```
echo "set bell-style none">> ~/.inputrc:
```

8.7.5 Messaggi di errore sulla console

L'impostazione di default da `/etc/syslog.conf` in Debian è per sovrascrivere lo schermo in console. Cambiate l'ultima parte di questo file.

Eseguite anche "dmesg -n1" per togliere i messaggi di errore su schermo.

Another place to look may be `/etc/syslog.conf`; check to see whether any messages are logged to a console device.

8.7.6 Riportare la console allo stato normale

Quando lo schermo impazzisce dopo "$ cat filebinario", (si raccomanda di digitare alla cieca):

```
$ reset
```

8.7.7 Convertire file di testo da DOS a Unix

Converte file di testo DOS (^M^J) in file Unix (^J).

```
# apt-get install sysutils
$ dos2unix dosfile
```

8.7.8 Convertire grossi files in files più piccoli

```
$ split -b 650m file   # divide il file in pezzetti di 650 MB
$ cat x* >largefile    # riunisce i files in un unico file
```

8.7.9 Parti di script per il piping

Gli script seguenti fanno cose utili come parti di una pipe.

```
xargs -n 1 command   # lancia i comandi per tutti gli oggetti da stdin
xargs -n 1 echo|     # divide lo spazio fra gli oggetti in righe
grep -e pattern|     # estrae le righe contenenti i modelli
cut -d: -f3 -|       # estrae il terzo campo separato da :
                     # (file passwd, ecc.)
col -bx |            # rimuove il backspace ed espande le tabs in spazi
expand -|            # espande le tabs
sort -u|             # trova e rimuove i doppioni
tr '\n' ' '|         # concatena le righe in una riga unica
tr '\r' ''|          # rimuove il CR (carriage return, a capo)
tr 'A-Z' 'a-z'|      # converte il maiuscolo in minuscolo
sed 's/^/# /'|       # commenta ogni riga
sed 's/\.ext//g'|    # rimuove .ext
sed  -n -e 2p|       # stampa una seconda riga
head -n 2 -|         # stampa le prime due righe
tail -n 2 -|         # stampa le ultime due righe
```

8.7.10 Cattura il testo / archivio delle ML da una pagina web.

Il seguente comando legge la pagina web e la copia in un file di testo. Molto utile quando si copiano delle configurazioni dalla rete.

```
$ lynx -dump http://www.remote-site.com/help-info.html >textfile
```

Se si tratta di un archivio di mailing list, usate "munpack" per ottenere i contenuti mime dal testo.

8.7.11 Il comando time

Mostra il tempo utilizzato dal processo.

```
# time df >/dev/null
real    0m0.035s       # tempo sull'orologio a muro (tempo realmente passato)
user    0m0.000s       # tempo in modalità utente
sys     0m0.020s       # tempo in modalità kernel
```

8.7.12 Il comando nice

nice (da GNU-shutils) per impostare il valore di nice al lancio del programma. renice (bsdutild) o top possono rimpostare nice. nice (20) significa un processo lentissimo. Not-nice (-20) significa super-user ed un processo molto veloce.

```
# nice --19 top                                              # molto nice
# nice --20 cdrecord -v -eject speed=2 dev=0,0 disk.img # molto veloce
```

8.7.13 Programmare gli eventi (cron, at)

Usate "cron" e "at" per programmare i task in Linux. Leggete "man at", "man 5 crontab" e "man 8 crontab".

Eseguite il comando "crontab" con il file crontab-file per impostare degli eventi con programmazione irregolare. Ecco un esempio di crontab-file:

```
# usa /bin/sh per eseguire i comandi, non importa cosa dice /etc/passwd
SHELL=/bin/sh
# invia ogni output a 'paul', non importa a chi appartiene crontab
MAILTO=paul
# Min Hour DayOfMonth Month DayOfWeek comando
# esecuzione alle 00:05, ogni giorno
5  0  *  * *   $HOME/bin/daily.job >> $HOME/tmp/out 2>&1
# esecuzione alle 14:15 il I giorno del mese -- output inviato a paul
15 14 1  * *   $HOME/bin/monthly
# esecuzione alle 22:00 i giorni feriali (Lun-Ven) (1-5), annoia Joe.
% per una nuova riga, l'ultimo % per cc:
0 22 *   * 1-5 mail -s "It's 10pm" joe%Joe,%%Where are your kids?%.%%
23 */2 1 2 *   echo "run 23 minutes after 0am, 2am, 4am ..., on Feb 1"
5  4 *   * sun echo "run at 04:05 every sunday"
# esecuzione alle 03:40 il I lunedi di ogni mese
40 3 1-7 * *   [ "$(date +%a)" == "Mon" ] && command -args
```

Usate il comando "at" per programmare gli eventi che accadono una sola volta:

```
$ echo 'command -args'| at 3:40 monday
```

8.7.14 CScambio di console con `screen`

Il programma `screen` vi permette di far girare più terminali virtuali, ciascuno con la propria shell interattiva, su un singolo terminale fisico, o su una finestra di emulazione di terminale. Anche se usate le consoles virtuali di Linux o finestre xterm multiple, vale comunque la pena di esplorare `screen` per la sua ricchezza di opzioni, che includono la memoria per lo scrollback, copia-e-incolla, registrazione dell'output, digraph entry e l'abilità di staccare una intera sessione di `screen` dal vostro terminale e di riattacarla in seguito. Se accedete frequentemente ad una macchina Linux da terminale remoto o tramite un prgramma di terminale VT100, `screen` renderà la vostra vita molto più semplice.

Una volta lanciato `screen`, tutto l'input della tastiera viene indirizzato alla finestra corrente, tranne che i comandi da tastiera, che di default è `^A`. Tutti i comandi di`screen` vengono dati digitando `^A` più una sigola lettera [più alcuni parametri]. Comandi utili:

```
^A ?  mostra una schermata di aiuto (mostra i comandi da tastiera) ^A c crea
una nuova finestra e passa ad essa ^A n va alla prossima finestra ^A p va alla
finestra precedente ^A 0 va alla finestra numero 0 ^A w mostra una lsta di
finestre ^A a manda un Ctrl-A alla finestra corrente come input da tastiera ^A
h scrive una copia della finetra corrente su file ^A H inizia/termina la
registrazione della finestra corrente su file ^A ^X blocca il terminale
(protetto da password) ^A d stacca la sessione dal terminale ^A DD stacca la
sessione ed esce
```

L'opzione "detach" di `screen` è molto potente. Supponiamo che siate connessi mediante una connessione dialup e che state conducendo una sessione di `screen` molto complessa, con editors ed altri programmi aperti su molte finestre. Inprovvisamente dovete abbandonare il terminale, ma non volete perdere tutto il lavoro allo staco della connessione. Basta digitare `^A d` per staccare la sessione, poi deconnettersi. (Oppure, più veloce ancora, digitate `^A DD` per dire a `screen` di staccare la sessione e deconnettersi da solo.) Quando vi riconnettete di nuovo, date il comando `screen -r` e `screen` magicamente riattaccherà tutte le finestre che avevate aperte.

Questo è solo un assaggio dei comandi e caratteristiche di `screen`. Se c'è qualcosa che volete che `screen` faccia, è probabile che lo sappia fare! "man screen" per i dettagli.

Nota: Se mentre state facendo girare `screen` notate che il backspace e/o Ctrl-H non funzionano appropriatamente, aprite `/etc/screenrc`, trovate la linea

```
bindkey -k kb stuff "\177"
```

e commentatela (cioè, aggiungete "#" come primo carattere).

8.7.15 Testare la rete: le basi

Installare i pacchetti `netkit-ping`, `traceroute`, `dnsutils`, `ipchains` (per Kernel 2.2), `iptables` (per Kernel 2.4) e `net-tools` e:

```
$ ping yahoo.com            # prova la connessione internet
$ traceroute yahoo.com      # traccia i pacchetti
$ ifconfig                  # prova la configurazione dell'host
$ route -n                  # prova la cofigurazione di routing
$ dig [@dns-server.com] host.dom [{a|mx|any}] |less
      # controlla i registri DNS di host.dom [@ dns-server.com]
      # per il record [{a|mx|any}]
$ ichains  -L -n |less      # controlla il packet filter (2.2 kernel)
$ iptables -L -n |less      # controlla il packet filter (2.4 kernel)
$ netstat -a | grep LISTEN  # scopre tutte le porte aperte
$ netstat -a                # scopre tutte le porte aperte
$ netstat -l --inet         # trova le porte in ascolto
$ netstat -ln --tcp         # trova le porte tcp in ascolto (numerico)
```

8.7.16 Eliminare la posta dallo spool locale

Per eliminare la posta:

```
# exim -q    # elimina la posta in attesa
# exim -qf   # elimina tutta la posta
# exim -qff  # elimina persino la posta "congelata"
```

"-qff" può essere migliore per lo script /etc/ppp/ip-up.d/exim .

8.7.17 Rimuovere la posta "congelata" (bloccata) dallo spool locale

Per rimuovere la posta bloccata dallo spool locale assieme al messaggio di errore nell'invio:

```
# exim -Mg `mailq | grep frozen | awk '{ print $3 }'`
```

8.7.18 Pulire il contenuto di un file

Nel caso al file possano accedere altri, non usare "rm".

```
$ :>file-da-ripulire
```

8.7.19 Files fantoccio

```
$ dd if=/dev/zero    of=nomefile bs=1k count=5 # 5KB di zeri
$ dd if=/dev/urandom of=nomefile bs=1m count=7 # 7MB di contenuto casuale
$ touch nomefile #  crea un file di 0B, od aggiorna mtime
```

8.7.20 chroot

Potete creare un sistema indipendente condividendo lo stesso kernel.

```
# mount /dev/hda1 /mnt/target
        # supponiamo che /dev/hda1 contenga 1 sistema
# chroot /mnt/target
... A questo punto è come se root fosse il contenuto di /dev/hda1
# mount proc /proc              # nel caso serva
... lanciate i comandi da dentro /dev/hda1
```

Ciò permette di caricare stable/testing/unstable su una sola macchina. In più, si può far girare un programma che utilizza molta memoria come dselect, su una macchina host, montando tramite NSF una macchina satellite sull'host ed entrare tramite chroot con permessi di lettura e scrittura.

Un sistema chroot può essere facilmente creato, in woody, con il comando `debootstrap`.

```
# mkdir potatochroot
# debootstrap potato potatochroot
# chroot potatochroot
# apt-setup # set-up /etc/apt/sources.list
```

8.7.21 Samba

Le basi per ottenere un file da Window$:

```
# mount -t smbfs -o username=myname,uid=my_uid,gid=my_gid \
        //server/share /mnt/smb  # monta il file Windows su Linux
# smbmount //server/share /mnt/smb \
        -o "username=myname,uid=my_uid,gid=my_gid"
# smbclient -L 192.168.1.2 # elenca le condivisioni su un computer
```

Capitolo 9

Perfezionare Debian

9.1 sudo - un ambiente di lavoro più sicuro

L'uso che faccio di sudo è fondamentalmente a protezione dalla mia stessa stupidità.Usare sudo è per me sempre meglio che lavorare sotto root. (Esiste un programma simile: super)

Installate "sudo" ed attivatelo impostando /etc/sudoers (`examples/`)

Questa configurazione permette ai membri del gruppo staff un accesso su qualsiasi comando da eseguire come root sotto sudo, ed ai membri di src un accesso a comandi selzionati eseguibili come root, sempre da sudo.

Il vantaggio di sudo è che richiede solo la user password, e che l'attività e monitorizzata. E' un modo carino per dare un minimo di autorità ad un amministratore alle prime armi. Per esempio:

```
$ sudo chown -R myself:mygrp .
```

Ovviamente se sapete la password di root (come la maggior parte degli utenti casalinghi sa) qualunque comando può essere eseguito come root dall'account dell'utente:

```
$ su -c "shutdown -h now"
Password:
```

So che dovrei restringere i privilegi dell'account admin in sudo ma, dopo tutto è il mio server casalingo, e non ne ho ancora sentito il bisogno.

Per un programma differente che permetta agli utenti ordinari di eseguire comandi con privilegi di root, vedere il pacchetto `super`.

9.2 Backup

```
# apt-get install changetrack
# vi changetrack.conf
```

9.3 Masterizzatori-writer

I masterizzatori con interfaccia ATAPI/IDE sono di recente diventati un'opzione molto popolare. Rappresentano un supporto valido per il backup e l'archivio di un sistema per l'utente casalingo (<$1 / 640 MB). Le informazioni più autorevoli le trovate a `/usr/share/doc/HOWTO/en-txt/CD-Writing-HOWTO.txt.gz` (LDP). Accertatevi che la versione sia successiva all'anno 2000.

9.3.1 Introduzione

Primo, ogni distruzione di dati sul masterizzatore porterà ad un danno irreparabile del CD. Prendete il masterizzatore con il buffer più grande possibile. Se i soldi non sono un problema, prendetene uno con interfaccia SCSI. Se potete scegliere quale interfaccia IDE connettere, prendete quello su PCI-bus (sulla scheda madre), piuttosto che su ISA-bus (schede SB16, ecc.).

Quando il masterizzatore è connesso su IDE, deve essere comandato dal driver IDE-SCSI, non dall'IDE-CD. Deve essere attivato anche il driver generico SCSI. Esistono due approcci per ottenere ciò, per i kernel delle moderne distribuzioni (Marzo 2001).

9.3.2 Approccio 1: (modules+lilo)

Aggiungete la riga seguente a /etc/lilo.conf se usate qualsiasi kernel standard di una delle maggiori distribuzioni (Debian, Redhat, Mandrake, ...). Se usate più opzioni, elencatele separate da spazi.

```
append="hdx=ide-scsi"
```

Qui la collocazione del masterizzatore, a cui si ha accesso tramite il driver ide-scsi, è indicata da `hdx`:

```
hdb      se slave sulla prima porta IDE.
hdc      se master sulla seconda porta IDE.
hdd      se slave sulla seconda porta IDE.
hde-hdh  se sulla porta IDE esterna.
```

Finita la configurazione, date il seguente comando per l'attivazione, come root:

```
# lilo
# shutdown -h now
```

9.3.3 Approccio 2: (ricompilare il kernel):

Debian usa make-kpkg per compilare il kernel. Usate il nuovo "–append_to_version" con make-kpkg per costruire kernel-images multiple. Usate la seguente configurazione tramite "make menuconfig":

- bzImage.
- Escludete il driver IDE-CD (Non obbligatorio, ma è più semplice fare così)
- Compilate ide-scsi e sg, direttamente nel kernel o come moduli.

9.3.4 Fasi post-configurazione

Il supporto del kernel per il masterizzatore può essere attivato al boot tramite:

```
# echo ide-scsi >>/etc/modules echo sg >>/etc/modules cd /dev; ln
# -sf scd0 cdrom
```

L'attivazione manuale può essere fatta tramite:

```
# modprobe ide-scsi modprobe sg
```

Dopo il reboot, potete controllare l'avvenuta installazione tramite:

```
$ dmseg|less # apt-get install cdrecord # cdrecord -scanbus
```

NdT: se volete montare anche i cd sul masterizzatore (per esempio avete solo questo sulla vostra macchina), non dimenticate di aggiungere al momento della compilazione del kernel il driver per il cdrom SCSI (sr) (anche se connesso su IDE), altrimenti potrete solo masterizzare e non leggere alcunchè.

9.3.5 Il file CD-image (bootabile)

Per creare un CD-ROM di files contenuti nella target-directory/ sotto forma di cd-image.raw (bootabile, con il formato Juliet TRANS.TBL attivato. Se non lo volete bootabile, togliete le opzioni -b e -c), inserite il boot floppy nel drive e,

```
# dd if=/dev/fd0 target-directory/boot.img
# mkisofs -r -V volume_id -b boot.img -c bootcatalog -J -T \
        -o cd-image.raw target_directory/
```

Una cosa carina è fare dei CD-ROM bootabili in DOS. Se una normale immagine DOS di un boot floppy è contenuta nel boot.img, il CD-ROM eseguirà il boot come se nel primo floppy drive (A:) ci fosse il floppy DOS. Più interessante potrebbe essere farlo con freeDOS.

Il file CD-image può essere ispezionato montandolo sul device loop.

```
# mount -t iso9660 -o ro,loop cd-image.raw /cdrom
# cd /cdrom
# mc
# umount /cdrom
```

9.3.6 Scrivere sul masterizzatore (R, R/W)

Primo test con (in caso di velocità doppia)

```
# nice --20 cdrecord -dummy speed=2 dev=0,0 disk.img
```

Se tutto va bene, scrivete sul CD-R con

```
# nice --20 cdrecord -v -eject speed=2 dev=0,0 disk.img
```

Oppure, sul CD-RWcon

```
# nice --20 cdrecord -v -eject blank=fast speed=2 dev=0,0 disk.img
```

9.3.7 Fare un'immagine di un CD

Alcuni CD-R e CD commerciali hanno dei settori spuri alla fine, cosa che rende impossibile la copia tramite dd (il CD di Windows 98 è uno di questi). Il pacchetto cdrecord ha il comando readcd. Usatelo per copiare qualsiasi CD in una file immagine. Se è un disco di dati, Montatelo ed eseguite "mount" per vedere le dimensioni reali. Dividete il numero ottenuto (in blocchi=1024byte) per 2 per avere il numero reale di settori del CD (2048 byte). Lanciate readcd con le opzioni ed usate l'immagine risultante per bruciare il CD-R/RW.

```
# readcd [target] [lun] [scsibusno] # select function 11
```

Per la maggior parte dei casi, impostate tutti e tre i parametri a 0. Talvolta il numero dei settori dati da readcd è un pò in eccesso! Utilizzate il numero ottenuto dal montaggio per i migliori risultati.

```
Il mio CD-R       = +2 sectors
MS Windows CD     = +1 sector, i.e., +2048 byte
```

9.3.8 Creare un CD Debian

Le informazioni più aggiornate le potete reperire dal Debian CD site. (http://cvs.debian.org/debian-cd/) Il software più aggiornato può essere reperito tramite CVS anonimo.

```
$ export CVSROOT=anonymous@cvs.debian.org:/cvs/debian-boot
$ cvs co debian-cd
```

Per la creazione del CD, date i seguenti comandi.

```
# apt-get install debian-cd
# cd /etc/debian-cd
# vim CONF.sh
$ . CONF.sh
$ make distclean
$ make status
$ make list COMPLETE=1 SIZELIMIT1=555000000 SRCSIZELIMIT=665000000
$ make official_images
 oppure se volete solo immagini binarie:
$ make bin-official_images
```

9.3.9 Backup del sistema su CD-R

Copiate i file di configurazione chiave e di dati sul CD-R. Usate questo script: backup (examples/)

9.3.10 Copiare CD musicali su CD-R

Non testato da me:

```
# apt-get get install cdrecord cdparanoia
# cdparanoia -s -B
# cdrecord dev=0,0,0 speed=2 -v -dao -eject defpregap=1 -audio *.wav
```

oppure,

```
# apt-get install cdrdao #disk at once
# cdrdao read-cd --device /dev/cdrom --driver generic-mmc \
        --paranoia-mode 3 my_cd # legge il cd
# cdrdao write --device /dev/cdrom --driver generic-mmc --speed 8 \
        my_cd # scive un nuovo CD
```

cdrdao crea una vera copia.. (senza buchi,.. ecc..)

9.4 I programmi X

Il server X è una macchina che mostra il sistema X window su schermo ed accetta input da tastiera. Il client X è una macchina su cui girano delle applicazioni compatibili con X window. Talvolta i due termini vengono invertiti, portando un pò di confusione. Per i dettagli fate riferimento al X(7), all' XWindow-User-HOWTO.txt.gz (LDP) e al mini/Remote-X-Apps.txt.gz (LDP).

9.4.1 Il server X

Per lanciarlo leggete "man 1 XFree86"

per (ri)configurare X4 in woody, lanciate:

```
# dpkg-reconfigure --p=low xserver-xfree86
```

Invocate il server X dalla console locale:

```
$ startx -- :<display> vtXX
eg.:
$ startx -- :1 vt8
... esempio per lanciarlo in vt8 connesso a localhost:1
```

Molti sono i modi per far accettare connessioni remote (lato applicazioni) al server (lato schermo). Il meccanismo della host list (xhost), il meccanismo della magic cookie (xauth), ed il port forwarding con ssh (shell sicura). Spiegherò i più semplici, xhost (insicuro) e ssh (sicuro).

9.4.2 Client X

Il programma client X (lato applicazione) ottiene il display a cui connettersi leggendo la variabile DISPLAY. Per esempio, prima di lanciare un programma client di X, eseguire uno dei seguenti comandi ottiene:

```
$ export DISPLAY=:0
        # default, la macchina locale con il primo schermo X
$ export DISPLAY=hostname.fulldomain.name:0.2
$ export DISPLAY=localhost:0
```

Dove,

- hostname:D.S significa schermo S sul display D dell'host hostname; il server X per questo display è in ascolto sulla porta TCP 6000+D.
- host/unix:D.S significa schermo S su display D dell'host host; il server è in ascolto sullo UNIX domain socket /tmp/.X11-unix/XD (in tal modo è raggiungibile solo dall'host).
- :D.S equivale a host/unix:D.S, dove host è l'hostname locale.

Questa impostazione può essere cambiata dando al client l'argomento -display hostname:0 al momento del lancio.

9.4.3 Connessione TCP/IP ad X

X11 non permette connessioni TCP/IP (remote) di default, a partire dalle recenti versioni di Debian. Per scoprire se è così sulla vostra macchina:

```
# find /etc/X11 -type f -print0 | xargs -0 grep nolisten
/etc/X11/xinit/xserverrc:exec /usr/bin/X11/X -dpi 100 -nolisten tcp
```

Rimuovete "-nolisten" per permettere le connessioni TCP/IP ad X.

9.4.4 Connessione X remota: Xhost

Xhost permette l'accesso basato sugli hostnames. Molto poco sicuro. Se utilizzato dietro un firewall con utenti fidati, il seguente comando disabilita il controllo dell'host e permette la connessione da ovunque:

```
$ xhost +
```

Si può riabilitare con:

```
$ xhost -
```

Non distingue fra utenti diversi sull'host remoto. In più gli hostnames (veri indirizzi) possono essere contraffatti. Cosa non buona se siete su una rete non sicura (tipo con una connessione telefonica PPP ad internet).

9.4.5 Connessione X remota: SSH

Questo è il modo semplice e sicuro di stabilire una connessione remota da un server locale ad un'applicazione su un server remoto. Lanciate il server locale su un sistema locale. Lanciate da esso x-term ed eseguite ssh per stabilire una connessione con il sito remoto. Eseguite, poi, i comandi delle applicazioni X del sito remoto.

```
localname @ localhost $ ssh -q -X -l loginname remotehost.domain
Password:
.....
loginname @ remotehost $ gimp &
```

9.4.6 Xterm

Tutto quello che avreste voluto sapere ma non avete mai osato chiedere su xterm presso: http://dickey.his.com/xterm/xterm.faq.html

9.4.7 Diventare root in X

Create $HOME/.xinitrc come segue:

```
#!/bin/sh
# X funziona da root quando si esegue su.
if [ -z "$XAUTHORITY" ]; then
        XAUTHORITY=$HOME/.Xauthority
        export XAUTHORITY
fi
# invoca lo script globale della sessione X
. /etc/X11/Xsession
```

e lanciate "su" da xterm.

9.5 SSH

SSH è il modo sicuro per connettersi ad internet. Io uso openSSH. [WOODY] usa SSH2. Attenzione alla migrazione.

9.5.1 Principi basilari

Per ssh, il pacchetto non-US deve essere disponibile per l'installazione.

```
# apt-get install ssh
```

La connessione ssh viene lanciata come segue.

```
$ ssh username@hostname.domain.ext $ ssh -1 username@hostname.domain.ext #
Force SSH version 1
```

Per l'utente, ssh è più intelligente di telnet (Non vi bombarda di with [^]).

9.5.2 Port forwarding - per il tunneling smtp/pop3

Per stabilire una pipe per connettere la porta 25 del server remoto con la port 4025 del localhost e la porta 110 del server remoto con la porta 4110 del localhost tramite ssh, eseguite dalla macchina locale:

```
# ssh -q -L 4025:remote-server:25 4110:remote-server:110 \
          username@remote-server
```

E' un modo sicuro di eseguire connessioni a server SMTP/POP3 su Internet.

9.5.3 Connettersi con meno passwords possibili - RSAAuthentication

[POTATO-SSH1] Potete evitare di ricordare una password per ogni sistema remoto usando la RSAAuthentication.

Impostate il contenuto di /etc/ssh/sshd_config sul sistema remoto a "RSAAuthentication yes" ed eseguite:

```
$ ssh-keygen $ cat .ssh/identity.pub | ssh user1@remote \ "cat -
>>.ssh/authorized_keys"
```

Potrete cambiare la passphrase in seguito con "ssh-keygen -p". Verificate le impostazioni provando la connessione. In caso di problemi, usate "ssh -v".

Potete aggiungere opzioni per limitare gli ingressi delle chiavi autorizzate sull'host. Vedere "man sshd" per i dettagli.

[WOODY-SSH2] Potete evitare di ricordare una password per ogni sistema remoto. Lanciate i seguenti comandi senza passphrase:

```
$ ssh-keygen -t rsa $ cat .ssh/id_rsa.pub | ssh user1@remote \ "cat -
<<.ssh/authorized_keys"
```

Potrete aggiungere/cambiare la passphrase in seguito con "ssh-keygen -p". Verificate le impostazioni provando la connessione. In caso di problemi, usate "ssh -v".

Potete aggiungere opzioni per limitare gli ingressi di authorized_keys2 sull'host e l'esecuzione di determinati comandi. Vedere "man sshd" per i dettagli.

[FIXME] SSH2 ha HostbasedAuthentication. Da Walter Hofmann walterh@gmx.de

```
C'erano due errori: 1) Bisogna passare ad HostbasedAuthentication sul client!
La pagina man è sbagliata, non avviene di default.  2) Bisogna impostare
PreferredAuthentications così che HostbasedAuthentication venga provata prima
di chiedere le passwords.
```

9.5.4 Problemi

In caso di problemi, controllate i permessi del file di configurazione e lanciate con l'opzione "-v".

Usate "-P" se non siete root ed avete problemi col firewall, per evitare di usare la porta del server (1:1023).

Dopo che l'amministratore del sistema remoto (potete anche essere voi stessi) ha combinato qualche casino, la connessione ssh può smettere di funzionare. In genere ciò capita dopo il cambiamento della host_key durante la manutenzione del sistema. Dopo essersi assicurati che è proprio questo il caso e che nessuno sta tentando di falsificare l'host remoto mediante qualche trucco geniale, potete riguadagnare la connessione rimuovendo la voce host_key dal ~/.ssh/known_hosts sulla macchina locale.

9.6 Programmi di gestione posta

La configurazione della posta si divide in tre fasi:

- MTA: exim
- MUA: mutt
- Utilities: procmail, fetchmail, mail,...

9.6.1 Mail transport agent (Exim)

Uare exim come mail transfer agent (MTA). Configurazione:

```
/etc/exim/exim.conf     "eximconfig" per crearlo e modificarlo
/etc/inetd.conf         decommentate smtp per lanciare exim come demone
/etc/email-addresses    Aggiunta una lista di indirizzi di partenza fasulli
verificare il filtro usando exim -brw, -bf, -bF, -bV, ... etc.
```

9.6.2 Raccolta di tutti gli indirizzi e-mail inesistenti (Exim)

In /etc/exim/exim.conf [WOODY], aggiungete nella sezione DIRECTORS alla fine, (dopo localuser: director) un indirizzatore (director) che raccolga tutti gli indirizzi che gli indirizzatori precedenti non sono in grado di risolvere (Secondo Miquel van Smoorenburg):

```
catchall:
  driver = smartuser
    new_address = webmaster@mydomain.com
```

Se si desidera avere maggiori dettagli per ogni dominio virtuale, ecc., aggiungete la seguente riga alla fine di exim.conf (Secondo me, non ancora testato):

```
*@yourdomain.com ${lookup{$1}lsearch*{/etc/email-addresses} \
        {$value}fail} T
```

Poi aggiungete in /etc/email-adresses la voce "*"

9.6.3 Utilità per la posta (fetchmail)

"fetchmail" viene eseguito in modalità demone e raccoglie la posta dagli account POP3 del vostro provider smistandoli nel sistema locale di posta. Configurazione:

```
/etc/init.d/fetchmail   vedere sotto per lo script
/etc/rc?.d/???fetchmail lancia update-rc.d fetchmail defaults 30
/etc/fetchmailrc        file di configurazione (chown 600, posseduto
da fetchmail)
/etc/init.d/fetchmail
```

Le informazioni su come lanciare fetchmail come demone da uno script init.d in potato risultano confuse (in woody ciò dovrebbe essere stato risolto). /etc/init.d/fetchmail and /etc/fetchmailrc (examples/)

Se i vostri header di posta sono infestati da ^M per colpa del mailer del vostro provider, aggiungete 'stripcr' alle opzioni in ~/.fetchmailrc:

```
options fetchall no keep stripcr
```

9.6.4 Utilità per la posta (procmail)

Procmail è il sistema locale di consegna/filtro della posta. Per ogni account si deve creare un file "~/.procmailrc". Per i dettagli vedere i miei esempi. (examples/)

9.6.5 Mail user agent (Mutt)

Usare mutt come mail user agent (MUA) in combinazione con vim. Personalizzatelo tramite "~/.muttrc":

```
# usa il modo visuale e "gq" per riformattare le citazioni
set editor="vim -c 'set tw=72 et ft=mail'"
#
# impostazione degli header, presa direttamente dal manuale
```

```
("Sven's Draconian header weeding")
#
ignore *
unignore from: date subject to cc
unignore user-agent x-mailer
hdr_order from subject to cc date user-agent x-mailer
auto_view application/msword
....
```

Aggiungete i seguenti a /etc/mailcap:

```
text/html; lynx -force_html %s; needsterminal;
application/msword; /usr/bin/antiword '%s'; copiousoutput;
description="Microsoft Word Text"; nametemplate=%s.doc
```

Così M$ Word .doc può essere letto (Non testato.)

Capitolo 10

Costruire un gateway con Debian

Debian offre gli strumenti per costruire un gateway multiuso, in grado di padroneggiare NAT, posta, DHCP, DNS cache, http proxy cache, CVS, NFS, e Samba per una LAN casalinga.

10.1 Configurazione di rete

10.1.1 Configurazione dell'host

Una LAN usa gli indirizzi IP per la seguente rete privata, per evitare interferenze con gli indirizzi IP provenienti da Internet.

```
Classe A: 10.0.0.0                    con mask 255.0.0.0
Classe B: 172.16.0.0 - 172.31.0.0     con mask 255.255.0.0
Classe C: 192.168.0.0 - 192.168.255.0 con mask 255.255.255.0
```

Per la configurazione IP, Debian usa `/etc/network/interfaces`.

Per esempio, se `eth0` si connette ad Internet tramite un indirizzo IP fornito da DHCP e `eth1` si connette alla LAN, `/etc/network/interfaces` sarà impostato come segue [WOODY]:

```
auto eth0
iface eth0 inet dhcp

auto eth1
iface eth1 inet static
address 192.168.1.1
```

```
network 192.168.1.0
netmask 255.255.255.0
broadcast 192.168.1.255
```

Date il seguente comando per aggiornare la configurazione di rete sulla base di un nuovo `/etc/network/interfaces`:

```
# /etc/init.d/networking restart
```

Ricorda: il file ”interfaces“ di ”woody“ non è compatibile con ”potato“.

Se il sistema utilizza una PC card di rete, l'impostazione di rete avviene tramite `/etc/pcmcia/network.opts` instead.

Se avete dubbi, controllate l'ouput dei seguenti comandi:

```
# ifconfig
# cat /proc/pci
# cat /proc/interrupts
# dmesg|more
```

10.1.2 IP-masquerade

Le macchine in una LAN possono accedere all'esterno attraverso un gateway su cui gira IP-masquerade (NAT).

```
# apt-get install ipmasq
```

Applicate le ”rules“ come da esempio per rafforzare la protezione di ”ipmasq“. Vedere `/usr/share/doc/ipmasq/examples/stronger/README`. Per la kernel-image-2.4 Debian, assicuratevi di caricare i moduli. Vedere 'Funzioni di rete' a pagina 70.

Per la kernel-image-2.2 Debian, modificate Z92timeouts.rul in /etc/masq/rules come segue:

```
# tcp, tcp-fin, udp
# 2hr, 10 sec, 160 sec - default
# 1 day, 10 min, 10 min - più lungo, per esempio
$IPCHAINS -M -S 86400 600 600
```

In più, se la rete è raggiunta tramite una PC card di rete, ipmasq deve essere lanciato da /etc/pcmcia/network.opts. (leggete /usr/share/doc/ipmasq/ipmasq.txt.gz)

10.1.3 Punti principali di una configurazione di rete

Programmi necessari per una installazione tipica:

```
# apt-get install nfs samba dhcpd dhcp-client bind squid procmail fetchmail
# apt-get install ssh cvs
```

Controllate, poi, i files seguenti:

```
/etc/init.d/dhcpd       (modificare per servire solo LAN = eth1)
/etc/host.allow         (ALL: 192.168.0.0/16 127.0.0.0/8) per NFS
/etc/exports            (Necessario per NFS)
/etc/bind/db.192.168.1  (aggiungere)
/etc/bind/db.lan        (aggiungere)
/etc/bind/named.conf    (modificare)
/etc/resolv.conf       (modificare)
/etc/hosts
/etc/dhcpd.conf         (modificare per LAN = eth1)
/etc/dhclient.conf      (modificare per forzare DNS locale)
/etc/samba/smb.conf
/etc/exim/exim.conf
/etc/mailname
/etc/aliases
/etc/squid.conf         (aggiungere tutti gli IP degli host della LAN
come autorizzati)
```

Bind crea un server locale DNS cache e modifica il DNS a localhost. Check `/etc/resolv.conf`:

```
nameserver 127.0.0.1
search lan.aokiconsulting.com
```

10.2 Gestione di connessioni multiple di rete

[DA CORREGGERE] Regole di routing: (da Phil Brutsche pbrutsch@tux.creighton.edu) Vedere Manuale iproute (http://ds9a.nl/2.4Routing/) per dettagli. Il traffic control (tc) potrebbe anche risultare interessante.

L'ambiente:

```
eth0: 192.168.1.2/24; gateway 192.168.1.1
eth1: 10.0.0.2/24; gateway 10.0.0.1
Nessun masquerading su questa macchina.
```

Alcune magie:

```
# ip rule add from 192.168.1.2 lookup 1
# ip rule add from 10.0.0.2 lookup 2
# ip route add to default via 10.0.0.1 metric 0
# ip route add to default via 192.168.1.1 metric 1
# ip route add table 1 to 192.168.1.0/24 via eth0
# ip route add table 1 to 10.0.0.2/24 via eth1
# ip route add table 1 to default via 192.168.1.1
# ip route add table 2 to 192.168.1.0/24 via eth0
# ip route add table 2 to 10.0.0.2/24 via eth1
# ip route add table 2 to default via 10.0.0.2
```

[DA CORREGGERE] Non l'ho mai fatto personalmente. Come impostare una connessione dialup come riserva di una connessione veloce?

Capitolo 11

Editor

11.1 Editor di testo

Linux offre molteplici alternative per gli editor di testo, fra queste troviamo:

- vim: Potente e leggero, ereditato da BSD. VI iMproved.
- emacs: L'editor finale, ultracompleto da GNU. L'originale da Richard M. Stallman.
- mcedit: Per neofiti, sempre da GNU. Uguale all'editor interno di mc.
- ae: Piccolo, default in potato. Da evitare.
- nano: Piccolo, default in woody. Emula pico.
- joe: Per i nostalgici di WordStar o TurboPascal.
- jed: Veloce, ricco di funzioni, basato su menu, con i comandi da tastiera di emacs.
- jove: Molto piccolo, con gli stessi comandi di emacs.
- nvi: Il nuovo vi. Totalmente compatibile con l'originale.

Usate "update-alternatives –config editor" per cambiare l'editor di default.

11.2 Emacs e Vim

11.2.1 Comandi utili per Vim

```
<F1>          Aiuto
```

```
<esc>          Ritorno al modo normale
V                    Modo Visuale
i                    Modo Inserimento
:                    Comandi da console
:set tw=72           Imposta larghezza testo a 72
<F11>          Modo copia testo
:r! date -R          Inserisce la data come da RFC-822
qa        Registra i tasti premuti nel registro a
@a        Esegue i comandi registrati dal registro a
```

`q` e `@` possono essere utilizzati per la registrazione e la riproduzione delle macro. Per esempio, per creare una macro che inserisca delle tag HTML in corsivo attorno ad una parola nella sede del cursore, si possono dare i seguenti comandi: `qii<i>^[ea</i>^[q` (dove `^[` è il tasto ESC). Poi, digitando `@i` all'inizio di una parola aggiungerà le tags **** e****.

11.2.2 Comandi utili per Emacs

```
<F1>           Aiuto
<F10>          Menu
C-u M-! date -R      Inserisce la data come da RFC-822
```

11.2.3 Lanciare l'editor

```
lancia l'editor:                emacs nomefile  vim nomefile
lancia un editor vi-compatibile:                vim -C
lancia un editor non vi-compatibile:            vim -N
lancia l'editor in modo compilatore di default:
                                emacs -q        vim -N -u NONE
```

11.2.4 Riassunto dei comandi per emacs e vim

```
esce:                           C-x C-c         :qa /:wq /:xa /:q!
Ritorno/modo comando:           C-g             <esc>
Indietro(sinistra):             C-b             h
Avanti(destra):                 C-f             l
Successivo(giù):                C-n             j
Precedente(su):                 C-p             k
Inizio riga(^):                 C-a             0
Fine riga($):                   C-e             $
```

Comandi multipli:	C-u nnn cmd	:count cmd
Comandi multipli:	M-numero cmd	
Salva File:	C-x C-f	:w file
Inizio buffer:	M-<	1G
Fine Buffer:	M->	G
avanti di una pagina:	C-v	^F
avanti di 1/2 pagina:		^D
avanti di una riga:		^E
indietro di una pagina:	M-v	^B
indietro di 1/2 pagina:		^U
indietro di una riga:		^Y
Scorri alla pagina superiore:	M-C-v	
cancella nel cursore:	C-d	x
cancella dal cursore alla fine della riga:	C-k	D
iCerca in avanti:	C-s	
iCerca indietro:	C-r	
cerca in avanti:	C-s enter	/
cerca indietro:	C-r enter	?
icerca regexp:	M-C-s	
icerca indietro regexp:	M-x isearch-backward-regexp	
cerca regexp:	M-C-s enter	/
cerca indietro regexp:	M-x isearch-backward-regexp enter	?
Aiuto:	C-h C-h	:help
Aiuto Apropos:	C-h a	
Aiuto comandi:	C-h b	:help [key]
Aiuto Info:	C-h i	
Aiuto modo Major:	C-h m	
Aiuto tutorial:	C-h t	:help howto
Annulla:	C-_	u
Ripeti:	C-f	^R
Marca la posizione del cursore:	C-@	m{a-zA-Z}
Scambia Marcatura e posizione:	C-x C-x	
vai al segno nel file corrente:		'{a-z}
vai al segno in un altro file:		'{A-Z}
copia regione (di testo):	M-w	{visual}y
elimina regione:	C-w	{visual}d
Copia e tieni il buffer:	C-y	p y
Copia e cancella il buffer:	M-y	p
Cambia una regione in maiuscolo:	C-x C-u	{visual}U
Cambia una regione in minuscolo:	C-x C-l	{visual}u

```
Inserisci carattere speciale:   C-q ottale/keystroke
                                    ^V decimale/keystroke
sostituisci:                    M-x replace-string      :%s/aaa/bbb/g
sostituisci regexp:             M-x replace-regexp      :%s/aaa/bbb/g
trova e sostituisci:            M-%                     :%s/aaa/bbb/gc
trova e sostituisci:            M-x query-replace
trova e sostituisci regexp:     M-x query-replace-regexp
Apri file:                      C-x C-f         :r file
Salva file:                     C-x C-s         :w
Salva tutti i buffers:          C-x s           :wa
Salva come...:                  C-x C-w file    :w file
Chiede un buffer specifico:     C-x b
Elenca i buffer:                C-x C-b         :buffers
Commuta in sola lettura:        C-x C-q         :set ro
Chiede ed elimina un buffer:    C-x k
Divide lo schermo in verticale: C-x 2           :split
Divide in orizzontale:          C-x 3           :vsplit (ver. 6)
Muove all'altra finestra:       C-x o           ^Wp
Cancella la finestra corrente:  C-x 0           :q
Cancella le/l'altra finestra(e):C-x 1           ^Wo
lancia una shell sullo sfondo:  M-x compile
uccide la shell sullo sfondo:   M-x kill-compilation
lancia make                                     :make Makefile
controlla i msg di errore:      C-x`            :echo errmsg
lancia una shell e registra:    M-x shell       :!script -a tmp
...pulisce BS, ...                                :!col -b <tmp >record
...salva/richiama una
registrazione:                  C-x C-w record  :r record
lancia una shell:               M-! sh          :sh
lancia un comando:              M-! cmd         :!cmd
lancia un comando ed inserisce: C-u M-! cmd     :r!cmd
lancia un filtro:               M-| file        {visual}:w file
lancia un filtro ed inserisce:  C-u M-| filter  {visual}:!filter
mostra le opzioni:                              :se[t] {option}?
riporta l'opzione al default                    :se[t] {option}&
resetta un'opzione booleana                     :se[t] no{option}
commuta un'opzioe booleana                      :se[t] inv{option}
a capo alla colonna 72                          :se tw=72
niente a capo                                   :se tw=0
autoindentazione                                :se ai
estende tabulazione                             :se et
specifica un commento (posta)                   :se comments=n:>,n:\|
```

```
Lancia GDB                        M-x gdb
descrivi il modo GDB              C-h m
salta una riga                    M-s
riga successiva                   M-n
salta una istruzione (stepi)      M-i
Finisci lo stack frame corrente C-c C-f
continua                          M-c
up arg frames                     M-u
down arg frames                   M-d
copia un numero a partire dal punto, inseriscilo alla fine
                                  C-x &
Imposta una cesura                C-x SPC
```

11.2.5 Configurare Vim

Per utilizzare tutte le caratteristiche di vim e la sintassi evidenziata, includete le linee seguenti in "~/.vimrc" oppure "/etc/vimrc":

```
set nocompatible
set nopaste
set pastetoggle=<f11>
syn on
```

Il modo "inserisci" evita che l'autoindentazione interferisca con le operazioni di taglia-e-incolla, se eseguite in terminale. E' molto di più di un semplice ":set noai".

11.2.6 Ctags

"apt-get install exuberant-ctags" e lanciate ctags sul file sorgente. Digitando ":tag function_name" in vim arrivate alla riga dove inizia "function_name". Funziona per C, C++, Java, Python e moltri altri linguaggi.

Emacs possiede la medesima funzionalità.

11.2.7 Convertire una schermata da "sintassi-evidenziata" in sorgente HTML

"so \$VIMRUNTIME/syntax/2html.vim" dal modo comando in vim converte il testo evidenziato in testo HTML. Salvate con ":w file.html" e ":q". Utile per codica in C, ecc.

11.2.8 Dividere lo schermo con `vim`

`vim` può aprire più files in un ambiente con uno schermo diviso in più finestre. Digitate ":help usr_08.txt" per i dettagli.

Per dividere lo schermo e mostrare files differenti, digitate al prompt di vi:

```
:split altro-file
:vsplit altro-file
```

o al prompt della shell:

```
$ vi -o file1.txt file2.txt   # Divide orizzontalmente
$ vi -O file1.txt file2.txt   # Divide verticalmente
```

risulterà un vi a molte finestre.

```
$ vimdiff file.txt~ file.txt          # controlla i cambiamenti recenti di file.txt
$ vimdiff file.en.sgml file.fr.sgml   # controlla i cambiamenti fra le traduzioni
$ gvimdiff file.txt~ file.txt         # in X
```

fornisce una buona visione con un file di backup. In SGML accoppia le tags, così fare una comparazione fra traduzioni è molto facile.

Movimenti speciali del cursore con i comandi CTRL-W:

```
CTRL-W +      ingrandisce una finestra
CTRL-W -      rimpicciolisce una finestra
CTRL-W h      va alla finestra di sinistra
CTRL-W j      va alla finestra inferiore
CTRL-W k      va alla finestra superiore
CTRL-W l      va alla finestra di destra
...
```

I controlli seguenti modificano lo scorrimento dello schermo:

```
:set scrollbind
:set noscrollbind
```

Capitolo 12

CVS

Leggete `/usr/share/doc/cvs/html-cvsclient`,`/usr/share/doc/cvs/html-info`,`/usr/share/doc/cvsbook` con lynx, od eseguite un ”info cvs“ e ”man cvs“ per informazioni dettagliate.

12.1 Installare un server CVS

Le impostazioni seguenti permettono il ”commit“ solo da parte di un membro del gruppo ”src“, e l’amministrazione del cvs solo da parte di un membro del gruppo ”staff“, riducendo così le possibilità di mettersi nei guai.

```
# cd /var/lib; umask 002 ; sudo mkdir cvs # [WOODY] FSH
# apt-get install cvs cvs-doc cvsbook
# export CVSROOT=/var/lib/cvs
# cd $CVSROOT
# chown root:src .          # mettete "staff" per aumentare le restrizioni,
                            # in caso di progetti allo stato iniziale.
# chmod 3775 .              # Se si usa "staff", utilizzate 2775
# cvs -d /var/lib/cvs init # è più sicuro specificare
                            # l’opzione -d esplicitamente!
# cd CVSROOT
# chown -R root:staff .
# chmod 2775 .
# touch val-tags
# chmod 664 history val-tags
# chown root:src history val-tags
```

12.2 Esempi di una sessione CVS

12.2.1 CVS anonimo (solo download)

```
$ export CVSROOT=:pserver:anonymous@cvs.qref.sf.net:/cvsroot/qref
$ cvs login
$ cvs -z3 co qref
```

12.2.2 Uso del server CVS locale

```
$ export CVSROOT=/var/lib/cvs
```

12.2.3 Uso di un pserver CVS remoto

Non molto sicuro, ma buono per un CVS anonimo:

```
$ export CVSROOT=:pserver:account@cvs.foobar.com:/var/lib/cvs
$ cvs login
```

12.2.4 Uso di un CVS remoto tramite ssh

Potete anche utilizzare l'utentificazione RSA ('Connettersi con meno passwords possibili - RSAAuthentication' a pagina 96), che elimina la necessità della richiesta di password.

```
$ export CVSROOT=:ext:account@cvs.foobar.com:/var/lib/cvs
```

oppure, per sourceforge:

```
$ export CVSROOT=:ext:account@cvs.qref.sf.net:/cvsroot/qref
```

12.2.5 Creare un nuovo archivio CVS

Per,

```
Oggetto          Valore             Significato
albero sorgente  ~/progetto-x       Tutti i codici sorgente
Nome progetto    progetto-x         Nome per questo progetto
Vendor Tag:      Main-branch        Tag per la branca intera
Release Tag:     Versione-originale Tag per una versione specifica
```

Quindi,

```
$ cd ~/progetto-x                 # entra nella directory sorgente
 ... crea un albero sorgente ...
$ cvs import -m "Start progetto-x" progetto-x Main-branch \
>   Versione-iniziale
$ cd ..; rm -R ~/progetto-x
```

12.2.6 Lavorare con CVS

Per richiamare e lavorare con le versioni locali del progetto-x con l'archivio CVS:

```
$ cd                # muove all'area di lavoro.
$ cvs co progetto-x # copia i sorgenti CVS all'archivio locale
$ cd progetto-x
 ... esegui cambiamenti al contenuto ...
$ cvs diff -u       # simile a diff -u repository/ local/
$ cvs ci -m "Describe change"   # salva i sorgenti locali nel CVS
$ vi newfile_added
$ cvs add newfile_added
$ cvs ci -m "Added newfile_added"
$ cvs up                        # inserisce l'ultima versione da CVS
$ cvs tag Release-1             # aggiunge la tag di versione
... esegui ulteriori modifiche ...
$ cvs tag -d Release-1          # rimuove la tag di versione
$ cvs ci -m "more comments"
$ cvs tag Release-1             # ri-aggiunge la tag di versione
$ cd                            # ritorna all'area di lavoro.
$ cvs co -r Release-initial -d old progetto-x
 ... riporta la versione originale alla più vecchia
$ cd old
$ cvs tag -b Release-initial-bugfixes # crea la tag di branca (-b)
 ... Ora si può lavorare sulla vecchia versione ...
$ cvs update -r Release-initial-bugfixes
 ... si fonde con le altre nella branca???
$ cvs update -j Release-initial-bugfixes
 ... si fonde nel tronco principale???
 ... Fissa i conflitti ...
$ cvs ci -m "merge Release-initial-bugfixes"
$ cd
$ tar -cvzf old-progetto-x.tar.gz old
```

```
                          # produce un archivio, -j per bz2
$ cvs release -d old      # rimuove i sorgenti locali (opzionale)
```

Alcune opzioni utili da ricordare (da usare subito dopo cvs):

```
-n      esecuzione secca, nessun effetto
-t      mostra messaggi sui passi dell'attività di cvs
```

12.2.7 Esportare i files da CVS

Per ottenere l'ultimissima versione da CVS, usate "tomorrow":

```
$ cvs ex -D tomorrow module_name
```

12.2.8 Amministrare CVS

Aggiungere un alias ad un progetto(server locale):

```
$ su - admin            # un membro del team
$ export CVSROOT=/var/lib/cvs ; cvs co CVSROOT/modules
$ cd CVSROOT
$ echo "px -a progetto-x" >>modules
$ cvs ci -m "Ora px è un alias per progetto-x"
$ cvs release -d .
$ exit                  # control-D per uscire da su
$ cvs co -d progettot px
       # progetto-x (alias:px) da CVS alla directory del progetto
$ cd project
 ... modifica il contenuto ...
```

12.3 Risoluzione dei problemi

12.3.1 Permessi dei file nel deposito

CVS non sovrascriverà il file corrente, ma lo rimpiazzerà con un altro. Quindi, *il permesso di scrittura nella directory deposito* è critico. Ogni volta che si crea un nuovo deposito, è bene lanciare i seguenti comandi per assicurarsi se detta condizione è necessaria.

```
# cd /var/lib/cvs
# chown -R root:src deposito
# chmod -R ug+rwX   deposito
# chmod    2775     deposito
        # se necessario, anche per le sottodirectory
```

12.3.2 Il bit d'esecuzione

Il bit d'esecuzione di un file viene mantenuto alla chiusura della sessione (check-out). Tutte le volte che appaiono problemi di permessi di esecuzione dei files, cambiate i permessi nel deposito CVS con il comando seguente.

```
# chmod ugo-x filename
```

12.4 Comandi CVS

Ecco i comandi CVS con le scorciatoie corrispondenti.

```
{add|ad|new} [-k kflag] [-m 'message'] files...
{admin|adm|rcs} [rcs-options] files...
{annotate|ann} [options] [files...]
{checkout|co|get} [options] modules...
{commit|ci|com}   [-lnR]  [-m  'log_message'  |  -f  file] \
        [-r revision] [files...]
{diff|di|dif} [-kl] [rcsdiff_options] [[-r rev1 | -D date1] \
        [-r rev2 |  -D date2]] [files...]
{export|ex|exp} [-flNn] -r rev|-D date [-d dir] [-k kflag] module...
{history|hi|his} [-report] [-flags] [-options args] [files...]
{import|im|imp} [-options] repository vendortag releasetag...
{login|logon|lgn}
{log|lo|rlog} [-l] rlog-options [files...]
{rdiff|patch|pa} [-flags] [-V vn] [-r t|-D d [-r t2|-D d2]] modules...
{release|re|rel} [-d] directories...
{remove|rm|delete} [-lR] [files...]
{rtag|rt|rfreeze} [-falnR]  [-b]  [-d]  [-r  tag  |  -D  date] \
         sym_bolic_tag modules...
{status|st|stat} [-lR] [-v] [files...]
{tag|ta|freeze} [-lR] [-F] [-b] [-d] [-r tag | -D date]  [-f] \
         sym_bolic_tag [files...]
{update|up|upd} [-AdflPpR] [-d] [-r tag|-D date] files...
```

Capitolo 13

Programmazione

Non usate ”test“ come nome di un file eseguibile di prova. ”test“ è un comando interno di shell.

13.1 Dove iniziare

Linux Programming Bible (John Goerzen/IDG books)

Documenti ed esempi in /usr/share/doc/<packages>

Molti documenti info più lunghi possono essere ottenuti rilegati da GNU (`http://www.gnu.org/`).

Le quattro sezioni a seguire contengono dei semplici scripts, in linguaggi differenti, per creare un file di testo contenente le informazioni di account, da aggiungere a /etc/passwd, utilizzando un ”batch processor“ come il programma ”newusers“. Ogni script richiede come input un file con le righe sotto forma di `nome cognome password`. (Le home directories reali di ciscun utente non vengono create con questi script.)

13.2 BASH

Learning the bash Shell, 2^ edizione (O'Reilly)

```
$ info bash
$ mc   /usr/share/doc/bash/examples/ /usr/share/doc/bash/
```

(Installate il pacchetto bash-doc per vedere i file di esempio.)

Esempio di programma breve (crea delle voci di account per "newusers" a partire da input standard):

```
#!/bin/bash
# (C) Osmu Aoki Sun Aug 26 16:53:55 UTC 2001 Public Domain
pid=1000;
while read n1 n2 n3 ; do
if [ ${n1:0:1} != "#" ]; then
let pid=$pid+1
echo ${n1}_${n2}:password:${pid}:${pid}:,,,/home/${n1}_${n2}:/bin/bash
fi
done
```

13.3 AWK

Effective awk Programming, 3^ edizione e *sed & awk*, 2^ edizione (O'Reilly)

```
$ man awk
$ info awk
```

Esempio di programma breve (crea delle voci di account per "newusers"):

```
#!/usr/bin/awk -f
# Script per creare un file utilizzabile con il comando 'newusers',
# a partire da un file che contiene user IDs e passwords sotto forma di:
# Nome Cognome password
# Copyright (c) KMSelf Sat Aug 25 20:47:38 PDT 2001
# Distributed under GNU GPL v 2, or at your option, any later version.
# This program is distributed WITHOUT ANY WARRANTY.

BEGIN {
    # Assign starting UID, GID
    if ( ARGC > 2 ) {
 startuid = ARGV[1]
 delete ARGV[1]
    }
    else {
        printf( "Usage:  newusers startUID file\n" \
   "...where startUID is the starting userid " \
```

```
    "to add, and file is \n" \
    "an input file in form firstname last name password\n" \
 )
 exit
    }

    infile = ARGV[1]
    printf( "Starting UID: %s\n\n", startuid )
}

/^#/ { next }

{
    ++record
    first = $1
    last = $2
    passwd = $3
    user= substr( tolower( first ), 1, 1 ) tolower( last )
    uid = startuid + record - 1
    gid = uid
    printf( "%s:%s:%d:%d:%s %s,,/home/%s:/bin/bash\n",  \
 user, passwd, uid, gid, first, last, user \
 )
}
```

13.4 PERL

Programming Perl, 3^ edizione (O'Reilly)

```
$ man perl
```

Esempio di programma breve (crea delle voci di account per "newusers"):

```
#!/usr/bin/perl
# (C) Osamu Aoki Sun Aug 26 16:53:55 UTC 2001 Public Domain
$pid=1000;
while (<STDIN>) {
 if (/^#/) { next;}
        chop;
        $pid++;
```

```
        ($n1, $n2, $n3) = split / /;
        print $n1,"_",$n2,":", $n3, ":",$pid,
                ":",$pid,",,,/home/",$n1,"_",$n2,":/bin/bash\n"
}
```

Installate Perl module <module name>:

```
# perl -MCPAN -e 'install <module name>'
```

13.5 PYTHON

Learning Python (O'Reilly). E' un interprete valido.

```
$ man python
```

Esempio di programma breve (crea delle voci di account per "newusers"):

```
#! /usr/bin/env python
import sys, string

# (C) Osamu Aoki Sun Aug 26 16:53:55 UTC 2001 Public Domain
# Ported from awk script by KMSelf Sat Aug 25 20:47:38 PDT 2001
# This program is distributed WITHOUT ANY WARRANTY.

def usages():
    print \
"Usage:  ", sys.argv[0], " start_UID [filename]\n" \
"\tstartUID is the starting userid to add.\n" \
"\tfilename is input file name. If not specified, standard input.\n\n" \
"Input file format:\n"\
"\tfirstname lastname password\n"
 return 1

def parsefile(startuid):
    #
    # main filtering
    #
    uid = startuid
    while 1:
```

```
        line = infile.readline()
        if not line:
            break
        if line[0] == '#':
            continue
        (first, last, passwd) = string.split(string.lower(line))
        # above crash with wrong # of parameters :-)
        user = first[0] + last
        gid = uid
        lineout = "%s:%s:%d:%d:%s %s,,/home/%s:/bin/bash\n" %  \
            (user, passwd, uid, gid, first, last, user)
        sys.stdout.write(lineout)
        +uid

if __name__ == '__main__':
    if len(sys.argv) == 1:
        usages()
    else:
        uid = int(sys.argv[1])
        #print "# UID start from: %d\n" % uid
        if len(sys.argv) > 1:
            infilename   = string.join(sys.argv[2:])
            infile = open(infilename, 'r')
            #print "# Read file from: %s\n\n" % infilename
        else:
            infile = sys.stdin
        parsefile(uid)
```

13.6 MAKE

Managing Projects with make, 2^ edizione (O'Reilly)

```
$ info make
```

Semplici variabili automatiche:

Regole di sintassi:

```
Obiettivo: [ Prerequisito ... ]
< TAB > command1
```

```
< TAB > -command2 # ignora errori
< TAB > @command3 # sopprime l'echoing
```

Ogni riga viene interpretata dalla shell dopo la sostituzione della variabile da parte di make. Usate "\" alla fine della riga per continuare a capo lo script. Usate "$$" per "$" per le variabili d'ambiente dello shell script.

Regola delle equivalenze implicite:

```
.c:   header.h == %  : %.c header.h
.o.c: header.h == %.c: %.o header.h
```

Variabili automatiche per le summenzionate regole:

```
foo.o: new1.c new2.c.c old1.c new3.c
$@ == foo.o                          (obiettivo)
$< == new1.c                         (il primo)
$? == new1.c new2.c new3.c           (i più recenti)
$^ == new1.c new2.c.c old1.c new3.c (tutti)
$* == '%' motivo corrispondente al motivo obiettivo.
```

Riferimenti delle variabili:

```
foo1 := bar    # Espansione unica
foo2  = bar    # Espansione ricorsiva
foo3 += bar    # Appendi
SRCS := $(wildcard *.c)
OBJS := $(foo:c=o)
OBJS := $(foo:%.c=%.o)
OBJS := $(patsubst %.c,%.o,$(foo))
DIRS  = $(dir directory/filename.ext) # Estrae la "directory"
$(notdir NAMES...), $(basename NAMES...), $(suffix NAMES...) ...
```

Per vedere le regole interne automatiche, lanciate "make -p -f/dev/null".

13.7 C

Kernighan & Ritchie, *The C Programming Language*, 2^ edizione (Prentice Hall). Per i riferimenti alla libreria delle funzioni GNU C:

```
# apt-get install glibc6-doc manpages-dev
```

Per i riferimenti alla libreria delle funzioni C, usate il comando "`info libc`". Per ciascuna funzione, tipo "printf", usate "`man 3 printf`".

13.7.1 Un semplice programma in C (gcc)

Un semplice esempio di compilazione del file "example.c", con una libreria "libm" in un eseguibile "run_example":

```
$ cat > example.c
#include <stdio.h>
#include <math.h>
#include <string.h>

int main(int argc, char **argv, char **envp){
 double x;
 char y[11];
 x=sqrt(argc+7.5);
 strncpy(y, argv[0], 10); /* prevent buffer overflow */
 y[10] = '\0'; /* fill to make sure string ends with '\0' */
 printf("%5i, %5.3f, %10s, %10s\n", argc, x, y, argv[1]);
 return 0;
}

$ gcc -Wall -g -o run_example example.c -lm
$ ./run_example
    1, 2.915, ./run_exam,     (null)
$ ./run_example 1234567890qwerty
    2, 3.082, ./run_exam, 1234567890qwerty
```

Qui, "-l*m*" è necessario per il link della libreria library "lib*m*" per sqrt(). La libreria vera è in "/lib" con il nome "libm.so.6", softlink a "libm-2.1.3.so".

Guardate l'ultimo parametro nel testo risultante. Ci sono più di 10 caratteri, anche se viene specificato "%10s".

L'uso di puntatori di funzioni che richiedano operazioni in memoria senza controlli sui loro "confini", tipo sprintf e strcpy, non è considerato sufficiente a prevenire gli effetti di exploit tipo buffer overflow, che annullano gli effetti di overrun. Utilizzate, invece, snprintf e strncpy.

13.7.2 Debugging (gdb)

GDB tutorial (http://www.dirac.org/linux/gdb)

Per il debugging di un programma compilato con l'opzione -g, usate gdb. Molti comandi possono essere abbreviati. L'espansione del comando mediante tab funziona come per la shell.

```
$ info gdb
...
$ gdb program
(gdb) b 1                # imposta il punto di interruzione alla riga 1
(gdb) run arg1 arg2 arg3 # lancia il programma
(gdb) next               # riga successiva
...
(gdb) step               # un passo avanti
...
(gdb) p parm             # stampa parm
...
(gdb) p parm=12          # imposta il valore di parm a 12
```

Anche i seguenti comandi possono essere utili.

- `ldd`: stampa le dipendenze delle librerie condivise
- `strace`: traccia le chiamate di sistema ed i segnali
- `ltrace`: traccia le chiamate alle librerie

Per il debugging da emacs, fate riferimento a 'Riassunto dei comandi per emacs e vim' a pagina 106.

13.7.3 Flex – una miglior... Lex

Al vostro programma dovete fornire una propria main() e yywrap(), altrimenti program.l apparirà come nell'esempio qui sotto, tentando la compilazione senza librerie (yywrap è una macro; "%option main" diventa implicitamente "%option noyywrap"):

```
%option main
%%
.|\n ECHO ;
%%
```

In alternativa, potete compilarlo con l'opzione -l*fl* al termine della riga di comando cc (tipo ATT-Lex con -l*l*). In questo caso nessuna %option è richiesta.

13.7.4 Bison – un Yacc migliore

Dovete fornire la vostra main() e yyerror(). Main() chiama yyparse() che chiama yylex(), normalmente creata con FleX.

```
%%

%%
```

13.7.5 Autoconf – disinstallare

SE avete ancora il sorgente e SE utilizza autoconf/automake e SE ricordate come l'avete configurato:

```
$ ./configure tutte-le-opzioni-che-avevate-dato
# make uninstall
```

13.8 SGML

SGML permette la gestione dei formati multipli dei documenti. Un sistema SGML semplice è Debiandoc, utilizzato qui. Richiede delle comversioni minori dai files di testo originali per i seguenti caratteri:

```
<   &lt;
>   &gt;
&   &amp;
©   &copy;
–   &ndash;
--   &mdash;
```

Per marcare una sezione come commento non stampabile, date:

```
<!-- Il commento va qui ... -->
```

Per marcare una sezione con un commento modificabile, date:

```
<![ %FIXME [ Il commento va qui ... ]]>
```

In SGML, la *prima definizione* di un'entità vince. Per esempio:

```
<!entity % qref "INCLUDE">
<![ %qref [ <!entity param "Data 1"> ]]>
<!entity param "Data 2">
&param;
```

Questa termina come "Data 1". Se la prima riga è, invece, "IGNORE", questa terminerà come "Data 2". La seconda riga è un'affermazione condizionale.

Per maggiori dettagli, fate un "`apt-get install debiandoc-sgml-doc`". Leggete, inoltre *DocBook The Definitive Guide*, di Walsh and Muellner (O'Reilly)

13.9 Creare pacchetti debian

Leggete la documentazione fornita nel pacchetto "packaging-manual" (Potato) o "debian-policy" (Woody).

Usate il comando dh_make dal pacchetto dh-make per creare un pacchetto di base. Poi, procedete secondo le istruzioni contenute in"`man dh_make`" Queste usano debhelper in debian/rules.

Un approccio alternativo è quello di usare deb-make dal pacchetto debmake. Non usa nessuno script debhelper e dipende esclusivamente dalla shell.

Per degli esempi di pacchetti con sorgenti multipli vedete "mc" (dpkg-source -x mc_4.5.54.dsc), che usa "sys-build.mk" di Adam Heath <doogie@debian.org> e "glibc" (dpkg-source -x glibc_2.2.4-1.dsc), che usa un altro sistema di Joel Klecker <espy@debian.org>.

Capitolo 14

GnuPG

Per informazioni dettagliate, vedere `/usr/share/doc/gnupg/README.gz`, oppure sfogliate `man gpg`.

14.1 Installazione di GnuPG

Leggete il "GNU privacy handbook" (in woody, `gnupg-doc`).

```
# gpg --gen-key                          # genera una nuova chiave
# gpg --gen-revoke mio_user_ID           # genera una chiave di revoca
per mio_user_ID
# host -l pgp.net | grep www|less        # cerca i pgp keyservers
```

Allo stato attuale buoni keyservers sono:

```
keyserver wwwkeys.eu.pgp.net
keyserver wwwkeys.pgp.net
```

Bisogna fare attenzione a non creare più di due sottochiavi, altrimenti i keyservers su pgp.net corromperanno la vostra chiave. In più, un solo keyserver può essere specificato in `$HOME/.gnupg/options`

Sfortunatamente i seguenti keyservers non sono più funzionanti:

```
keyserver search.keyserver.net
keyserver pgp.ai.mit.edu
```

14.2 Uso di GnuPG

Gestione dei file:

```
$ gpg  [options]  command  [args]
$ gpg {--armor|-a} {--sign|-s} file # firma un file in un file testo
                                    # file.asc
$ gpg --clearsign file              # firma un file in chiaro
$ gpg --clearsign --not-dash-escaped patchfile  # firma un file senza
                                                # patchfile
$ gpg --verify file                 # verifica un file firmato in chiaro
$ gpg -o file.sig {-b|--detach-sign} file    # crea una firma staccata
$ gpg --verify file.sig file        # verifica un file tramite file.sig
$ gpg -o crypt_file {--recipient|-r} nome {--encrypt|-e} file
       # Criptatura mediante chiave pubblica destinata a
$ gpg -o crypt_file {--symmetric|-c} file # criptatura simmetrica
$ gpg -o file --decrypt crypt_file  # decrittatura
```

14.3 Gestione di GnuPG

Gestione delle chiavi:

```
$ gpg --edit-key user_ID              # "help" per aiuto, interattivo
$ gpg -o file --exports               # esporta tutte le chiavi in un file
$ gpg --imports file                  # importa tutte le chiavi da file
$ gpg --send-keys user_ID             # invia le chiavi dello user_ID specificato
                                      # al keyserver
$ gpg --recv-keys user_ID             # recupera la chiave dello user_ID dal
                                      # keyserver
$ gpg --list-keys user_ID             # elenca le chiavi dello user_ID
$ gpg --list-sigs user_ID             # elenca le firme dello user_ID
$ gpg --check-sigs user_ID            # verifica le firme dello user_ID
$ gpg --fingerprint user_ID           # verifica il fingerprint dello user_ID
$ gpg --list-sigs | grep '^sig' | grep '[User id not found]' \
  | awk '{print $2}' | sort -u | xargs gpg --recv-keys # prende le chiavi
                                                       # sconosciute
  # aggiorna le chiavi per tutte le firme sconosciute.
```

Codice di confidenza:

```
-         Nessun codice assegnato / non ancora calcolato.
e         Il calcolo della confidenza è fallito.
q         Informazioni insufficienti per il calcolo.
n         Non confidare mai su detta chiave.
m         Confidenza parziale.
f         Confidenza totale.
u         Confidenza finale.
```

14.4 Uso con Mutt

Aggiungete le righe seguenti a ~/muttrc per evitare che uno GnuPG troppo lento venga lanciato automaticamente, ma solo quando necessario.

```
macro index S ":toggle pgp_verify_sig\n"
set pgp_verify_sig=no
```

Capitolo 15

Supporto per Debian

Per ottenere aiuto, consigli e supporto per Debian si possono seguire i riferimenti seguenti. Prima di chiedere aiuto (magari urlando) in lista, sarebbe bene fare del proprio meglio per utilizzare le fonti che uno ha a disposizione :) (Conoscete tutti l'acronimo RTFM, vero?)

Notate che potete accedere ad una buona fetta della documentazione presente sul vostro sistema tramite un banale browser WWW, tramite i comandi "dwww" o "dhelp", che trovate nei rispettivi pacchetti.

15.1 Riferimenti

I seguenti riferimenti sono disponibili per Debian e Linux (SO affini a Unix) in generale. Qualora trovaste informazioni contrastanti, fidatevi sempre della fonte primaria di informazioni, e non delle fonti secondarie come, ad esempio, questo documento.

- Manuale di installazione (fonte primaria)
  - Da leggere prima di installare ed aggiornare.
  - Web: `http://www.debian.org/releases/stable/installmanual`
  - Web: `http://www.debian.org/releases/testing/installmanual` (lavoro in corso)
  - Pacchetto: `install-doc`
  - File: `/usr/share/doc/install-doc/index.html`
- Note di rilascio (primaria)

  - Da leggere obbligatoriamente prima di installare ed aggiornare, anche se pensate di sapere tutto.
  - Web: http://www.debian.org/releases/stable/releasenotes
  - Web: http://www.debian.org/releases/testing/releasenotes (lavoro in corso)
  - Pacchetto: install-doc
  - File: /usr/share/doc/install-doc/upgrade-i386.html[FIXME]

- FAQ (secondaria)
  - Domande più frequenti (non aggiornatissime)
  - Web: http://www.debian.org/doc/manuals/debian-faq/
  - Pacchetto: doc-debian
  - File: /usr/share/doc/debian/FAQ/index.html

- APT HOWTO (secondaria)
  - Guida utente dettagliata per il sistema di gestione dei pacchetti in Debian (woody)
  - Web: http://www.debian.org/doc/manuals/apt-howto/
  - Pacchetti: apt-howto,
  - File: /usr/share/doc/apt-howto

- dselect Documentazione per principianti (secondaria)
  - Corso per dselect
  - Web: http://www.debian.org/releases/woody/i386/dselect-beginner
  - Pacchetto: install-doc,
  - File: /usr/share/doc/install-doc/[FIXME]

- Debian Policy Manual (primaria)
  - Testo di riferimento Debian.
  - Web: http://www.debian.org/doc/debian-policy/

- Debian Developer's Reference (primaria)
  - Informazioni base per sviluppatori.
  - Leggetelo una volta per noi comuni mortali (che non lo leggeremo mai).
  - Web: http://www.debian.org/doc/manuals/developers-reference/

- Nuova guida del manutentore Debian (primaria)
  - Guida pratica per sviluppatori.
  - Corso di "pacchettizzazione" per comuni mortali.
  - http://www.debian.org/doc/manuals/maint-guide/
- Manuale per la produzione dei pacchetti (potato)
  - Pacchetto: packaging-manual in potato.
- Pagine man (primaria)
  - man <package-name>
- Pagine info (primaria)
  - info <package-name>
- Documenti specifici dei pacchetti (primaria)
  - Li trovate in /usr/share/doc/<package-name>
- LDP: Linux Documentation Project (secondaria)
  - HOWTO e mini-HOWTO per Linux in genere
  - Web: http://www.tldp.org/
  - Pacchetto: doc-linux
  - File: /usr/share/doc/HOWTO/
- DDP: Debian Documentation Project (secondaria)
  - Manuali specifici per Debian
  - Web: http://www.debian.org/doc/
- Debian Developers' Corner (secondaria)
  - Informazioni chiave per gli sviluppatori Debian
  - Approfondimento per l'utente finale
  - Web: http://www.debian.org/devel/
- Codice sorgente (assolutamente primaria)
  - Nessuno può contestarlo :-)
  - Scaricare il codice seguendo 'Il codice sorgente' a pagina 11

15.2 Trovare il significato di una parola

Molti termini usati in Debian sono criptici. In più vengono utilizzati anche molti acronimi. Il seguente comando risolverà molti dubbi:

```
$ dict <metti-qui-un-termine-sconosciuto>
```

15.3 Il sistema Debian di tracciamento dei bachi

La distribuzione Debian ha un bug tracking system (BTS) (http://bugs.debian.org/) che archivia i dettagli dei bachi riportati dagli utenti e dagli sviluppatori. Ad ogni baco viene assegnato un numero, e viene mantenuto archiviato finchè non è segnato come risolto.

Prima di segnalare un baco, dovreste controllare se qualcun altro non lo ha già fatto. La lista dei bachi più importanti è disponibile in World Wide Web (http://bugs.debian.org/) and altrove (http://www.debian.org/Bugs/Access). Vedere anche 'Controllo dei bachi in Debian' a pagina 55.

Il metodo per riportare i bachi è descritto in http://www.debian.org/Bugs/Reporting

15.4 Mailing lists

Per tenervi aggiornati, leggete almeno debian-devel-announce (Inglese, sola lettura, basso traffico)

Le liste di principale interesse per l'utente Debian sono la debian-user (Inglese, aperta, alto traffico) e la debian-italian per (come avrete capito) l'italiano.

Per informazioni e dettagli sulla iscrizione vedere http://lists.debian.org/. Controllate sempre gli archivi prima di porre domande, ed attenetevi agli standard di etichetta.

15.5 Internet Relay Chat

Debian ha un canale IRC dedicato al supporto ed all'aiuto degli utenti, situato sul network Open Projects IRC, dedicato proprio al fornire collaborazione e alla condivisione di risorse per la comunità Open Source. Per accedervi puntate il vostro client IRC su irc.openprojects.net e unitevi a #debian.

Seguite le regole del canale, rispettando pienamente gli altri utenti. Se volete maggiori informazioni su Open Projects visitatene il sito (http://www.openprojects.net).

15.6 Motori di ricerca

Molti sono i motori di ricerca che cercano anche tra la documentazione relativa a Debian:

- La pagina Debian per la ricerca sul WWW (http://search.debian.org/).
- Google (http://www.google.com/): includere "site:debian.org" come termine di ricerca.
- Google Groups (http://groups.google.com/): motore di ricerca per newsgroups. Includere "linux.debian.user" come termine di ricerca.
- AltaVista (http://www.altavista.com/)

Per esempio, la ricerca su una stringa tipo "cgi-perl" fornisce una spiegazione molto più dettagliata su questo pacchetto rispetto alla breve descrizione del suo file di controllo. Vedere per esempio 'Controllo dei bachi in Debian' a pagina 55

15.7 Siti Web

Alcuni siti popolari con informazioni generali di supporto.

- Debian planet (http://www.debianplanet.org/)
- Linux.com (http://linux.com/)
- Matt Chapman's pages (Unix Guide) (http://www.belgarath.demon.co.uk/)

Di seguito alcuni indirizzi casuali da me raccolti per argomenti specifici.

- Adrian Bunk's packages to run kernel 2.4.x on potato (http://www.fs.tum.de/~bunk/kernel-24.html)
- Linux on Laptops (http://www.linux-laptop.net/)
- Xterm FAQ (http://dickey.his.com/xterm/xterm.faq.html)
- EXT3 File System mini-HOWTO (http://www.symonds.net/~rajesh/howto/ext3/index.html)
- Large File Support in Linux (http://www.suse.de/~aj/linux_lfs.html)
- LNX-BBC (Business-card-sized boot CD project) (http://www.lnx-bbc.org/)

- Linux info by Karsten Self (Partitioning, backup, browsers...) (http://kmself.home.netcom.com/Linux/)
- Backup info HOWTO by Alvin Oga (http://www.Linux-Backup.net/)
- Security info HOWTO by Alvin Oga (http://www.Linux-Sec.net/)
- Varie fonti NON-UFFICIALI per APT (http://www.internatif.org/bortzmeyer/debian/apt-sources/)
- Laptop Ethernet Configuration (http://www.orthogony.com/gjw/lap/lap-ether-intro.html)

Appendice A

Appendice

A.1 Autori

La guida Debian è stata iniziata da Osamu Aoki <debian@aokiconsulting.com> come memo personale di installazione e successivamente è stata chiamata ”Quick Reference . . . “. Buona parte del contenuto proviene dagli archivi della mailing list debian-user. Anche la ”Debian – Installation Manual“ e le ”Debian – Release Notes“ sono state utilizzate.

Seguendo il consiglio di Josip Rodin, che è molto attivo al ”Debian Documentation Project (DDP)“ ed è anche l'attuale responsabile della ”Debian FAQ“, questo documento è stato rinominato ”Debian reference“ ed è stato fuso con parecchi capitoli provenienti dalla ”Debian FAQ“.

Il documento originale è stato modificato e tradotto dalle seguenti persone:

- Per il documento originale inglese ”Quick Reference. . . “
  - Osamu Aoki <debian@aokiconsulting.com>
- Per la correzione delle bozze e la riscrittura inglese
  - David Sewell <dsewell@virginia.edu>
  - Brian Nelson <nelson@bignachos.com>
  - Daniel Webb <webb@robust.colorado.edu>
- Traduzione francese
  - Guillaume Erbs <gerbs@free.fr>
  - Rénald Casagraude <rcasagraude@interfaces.fr>

  - Jean-Pierre Delange <delange@imaginet.fr>
  - Daniel Desages <daniel@desages.com>

- Traduzione italiana

  - Davide Di Lazzaro <mc0315@mclink.it>

QREF è l'abbreviazione del titolo originale del documento, "Quick Reference..." ed è anche il nome del progetto presso qref.sourceforge.net.

Gran parte del contenuto del capitolo 'Nozioni fondamentali della Debian' a pagina 5 proviene dalla "Debian FAQ" (Marzo 2002):

- 5 Gli archivi Debian "ftparchives.sgml" (tutto il capitolo)
- 6 Il sistema di gestione dei pacchetti Debian "pkg_basics.sgml" (tutto il capitolo)
- 7 Panoramica degli strumenti di gestione dei pacchetti "pkgtools.sgml" (tutto il capitolo)
- 8 Aggiornare un sistema Debian "uptodate.sgml" (tutto il capitolo)
- 9 Debian ed il kernel "kernel.sgml" (tutto il capitolo)
- 10 Personalizzare l'installazione del vostro Debian GNU/Linux "customizing.sgml" (parte del capitolo)

Queste sezioni della "Debian FAQ" sono state incluse nel documento dopo una importante riorganizzazione, per riflettere i recenti cambiamenti del sistema Debian. Il contenuto di questo documento è più recente.

La "Debian FAQ" originale era stata fatta e mantenuta da J.H.M. Dassen (Ray) e Chuck Stickelman. Gli autori della "Debian FAQ" riveduta e corretta sono Susan G. Kleinmann e Sven Rudolph. Dopo di loro, la "Debian FAQ" è stata mantenuta da Santiago Vila. Il responsabile attuale è Josip Rodin.

Parte delle informazioni per la "Debian FAQ" provengono:

- L'annuncio del rilascio della Debian-1.1, di Bruce Perens (`http://www.perens.com/`).
- La Linux FAQ, di Ian Jackson (`http://www.chiark.greenend.org.uk/~ijackson/`).
- Debian Mailing List Archives (`http://lists.debian.org/`),
- il "dpkg programmers'" manual ed il "Debian Policy manual" (vedere 'Riferimenti' a pagina 131)

- tanti sviluppatori, volontari e beta testers, e la
- memoria traballante dei suoi autori. :-)

Gli autori desiderano ringraziare tutti coloro che hanno dato il loro contributo per rendere questo documento possibile.

A.2 Garanzie

Dato che non sono un esperto, non pretendo di possedere la verità assoluta su Debian o Linux in generale. Le considerazioni sulla sicurezza che faccio sono solamente applicabili per un uso casalingo.

Questo documento non rimpiazza alcuna guida di riferimento.

Non esistono garanzie. Tutti i marchi registrati appartengono ai rispettivi proprietari.

A.3 Feedback

Commenti ed aggiunte a questo documento sono sempre i benvenuti. Mandate una mail a Osamu Aoki (http://www.aokiconsulting.com/) <debian@aokiconsulting.com> in inglese, oppure ai traduttori delle rispettive lingue.

Dato che non sono di lingua madre inglese, qualsiasi correzione della grammatica è la benvenuta.

La cosa migliore è un diff per SGML, ma anche per la versione testo è accettabile.

A.4 Reperibilità

L'ultima versione ufficiale è http://www.debian.org/doc/manuals/debian-reference/ ed il progetto è ospitato presso http://qref.sourceforge.net/.

Il testo è disponibile per essere scaricato in formato testo, HTML, PDF e PostScript.

I files SGML originali usati per creare il documento sono disponibili in CVS presso: :pserver:anonymous@cvs.qr oppure http://qref.sourceforge.net/Debian/qref.tar.gz.

A.5 Formato del documento

Il documento è stato scritto utilizzando il DebianDoc SGML DTD (riscritto da LinuxDoc SGML). Il sistema DebianDoc SGML ci permette di creare dei files in formati diversi, a partire da un unico sorgente, ovvero potete leggere questo documento come HTML, testo, TeX DVI, PostScript, PDF, o GNU info.

Le utilità per la conversione da DebianDoc SGML sono disponibili nel pacchetto Debian `debiandoc-sgml`.

A.6 Il labirinto di Debian

Il sistema Debian è una piattaforma molto potente per l'utilizzo in rete. Tuttavia, imparare a padroneggiare tutti i suoi strumenti non è così semplice. Un buon esempio di quanto affermo può essere riscontrato nel configurare una stampante.

Esiste una mappa di questo sistema, completa e dettagliata, chiamata "CODICE SORGENTE". E' una mappa molto accurata, ma altrettanto difficile da comprendere. Altre fonti di informazione sono gli HOWTO e mini-HOWTO, più facili da comprendere, che però tendono a dare forse troppi dettagli, perdendo di vista l'insieme. Mi capita talvolta, quando ho bisogno di trovare dei comandi da usare, di avere dei problemi a trovare la sezione giusta in un lungo HOWTO.

Proprio per viaggiare attraverso i meandri della configurazione di un sistema Linux, iniziai a scrivere delle semplici note in formato testo, da utilizzare come riferimenti rapidi. L'elenco delle note crebbe sempre più con il passare del tempo, e nel frattempo imparai debiandoc. Il risultato finale è *La guida Debian*.

Ho fatto una scelta ponderata di non spiegare qualcosa se potete trovarla in una pagina man, info, oppure in un HOWTO. Ho anche tentato di non perdermi in troppe spiegazioni, ma di dare informazioni più dirette tramite delle precise sequenze di comandi o script di esempio. Il pubblico a cui mi rivolgo deve essere in grado di leggere uno shell script.

Il principio guida è *Keep it short and simple* (KISS), ovvero sii semplice e conciso.

Non essendo un esperto di Linux, non pretendo di essere onniscente su Debian o Linux. Le mie considerazioni sulla sicurezza sono applicabili solo per un utilizzo casalingo. Non possono in alcun modo rimpiazzare guide con molta maggiore autorità nel campo.

Il documento originale è in Debiandoc SGML, con ogni capitolo organizzato come un file separato. Le conversioni dall'originale SGML in html, testo, ps e pdf vengono eseguite tramite lo script make.

Molte idee sono state prese dalla mailing list "debian-user".