

The Debian GNU/Linux FAQ

Gli autori sono elencati in Autori delle FAQ di Debian

versione 8.1, 28 August 2016

Estratto

Questo documento risponde alle domande fatte spesso su Debian GNU/Linux.

Avviso di Copyright

Copyright © 1996-2016 by Software in the Public Interest, portions copyright © 2004, 2005, 2006 Kamaraju Kusumanchi

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	Definizioni e introduzione	1
1.1	Cosa sono queste FAQ?	1
1.2	Cosa è Debian GNU/Linux?	1
1.3	OK, ora so cosa è Debian... cosa è Linux?!	2
1.4	Debian fa solo GNU/Linux?	2
1.5	Qual è la differenza tra Debian GNU/Linux e le altre distribuzioni Linux? Perché scegliere Debian invece di qualche altra distribuzione?	2
1.6	Come si rapporta il progetto Debian con il progetto GNU della Free Software Foundation?	3
1.7	Come si pronuncia Debian e cosa significa questa parola?	3
2	Ottenere e installare Debian GNU/Linux	5
2.1	Qual è la versione più recente di Debian?	5
2.2	Esistono aggiornamenti dei pacchetti in "stable"?	5
2.3	Dove/Come si possono ottenere le immagini di installazione di Debian?	5
2.4	Come si installa Debian da CD-ROM?	6
2.5	Perché il CD-ROM ufficiale del rilascio "stable" contiene collegamenti simbolici "frozen" e "unstable"? Questo CD doveva contenere solo "stable"!	6
2.6	È possibile ottenere e installare Debian direttamente da un sito Internet remoto?	6
2.7	Esistono strategie alternative per avviare l'installatore di sistema?	6
3	Scegliere una distribuzione Debian	7
3.1	Quale distribuzione Debian (stable/testing/unstable) è meglio per me?	7
3.1.1	È stato suggerito di installare stable, ma in essa l'hardware pincopallino non viene rilevato o non funziona. Cosa fare?	7
3.1.2	Nelle differenti distribuzioni ci sono versioni diverse dei pacchetti?	8
3.1.3	Le distribuzioni stable contengono pacchetti veramente datati. Basta guardare Kde, Gnome, Xorg o persino il kernel: sono molto vecchi. Perché?	8
3.1.4	Se si decidesse di passare ad un'altra distribuzione, sarebbe possibile farlo?	8
3.1.5	Potete dirmi se installare stable, testing o unstable?	8
3.1.6	È stato detto che testing si può rompere. Cosa significa?	8
3.1.7	Come mai testing può essere danneggiata per mesi? Le soluzioni introdotte in unstable non passano direttamente in testing?	9
3.1.8	Dal punto di vista di un amministratore, quale distribuzione richiede più attenzioni?	9
3.1.9	Cosa succede quando viene fatto un nuovo rilascio?	9
3.1.10	Ho un desktop/cluster in funzione con installata Debian. Come fare a sapere quale distribuzione è in esecuzione?	10

3.1.11	I am currently tracking stable. Can I change to testing or unstable? If so, how?	10
3.1.12	Attualmente sto usando testing (stretch). Cosa succederà quando verrà fatto un rilascio? Continuerò ad usare testing o la macchina userà la nuova distribuzione stabile?	11
3.1.13	Sono ancora confuso. Cosa hai detto che devo installare?	11
3.2	But what about Knoppix, Linux Mint Debian Edition, Ubuntu, and others?	11
3.2.1	I know that Knoppix/Linux Mint Debian Edition/Ubuntu/... is Debian-based. So after installing it on the hard disk, can I use 'apt' package tools on it?	11
3.2.2	I installed Knoppix/Linux Mint Debian Edition/Ubuntu/... on my hard disk. Now I have a problem. What should I do?	12
3.2.3	I'm using Knoppix/LMDE/Ubuntu/... and now I want to use Debian. How do I migrate?	12
4	Problemi di compatibilità	13
4.1	Su quali architetture o sistemi hardware funziona Debian?	13
4.2	Quali kernel può eseguire Debian GNU/Linux?	14
4.3	Quanto è compatibile Debian con le altre distribuzioni Linux?	14
4.4	Quanto è compatibile Debian con altri sistemi Unix a livello di codice sorgente?	14
4.5	Si possono usare i pacchetti Debian (file ".deb") sul proprio sistema Linux RedHat/Slackware/...? Si possono usare i pacchetti RedHat (file ".rpm") sul proprio sistema Debian GNU/Linux?	15
4.6	Come installare un programma non-Debian?	15
4.7	Perché non posso compilare programmi che richiedono libtermcap?	15
4.8	Perché non posso installare AccelX?	16
5	Software disponibile nel sistema Debian	17
5.1	Quali tipi di applicazioni e software di sviluppo sono disponibili per Debian GNU/Linux?	17
5.2	Chi ha scritto tutto questo software?	17
5.3	Come si può ottenere una lista aggiornata dei programmi che sono stati impacchettati per Debian?	17
5.4	Come si può installare un ambiente di sviluppo per compilare pacchetti?	18
5.5	Cosa manca da Debian GNU/Linux?	18
5.6	Perché ricevo il messaggio "ld: cannot find -lpippo" quando compilo i programmi? Perché non c'è alcun file libpippo.so nei pacchetti delle librerie Debian?	18
5.7	Debian supporta Java? (E come?)	18
5.8	How can I check that I am using a Debian system, and what version it is?	19
5.9	Come supporta Debian le lingue diverse dall'inglese?	19
5.10	Dove è ezmlm/djbdns/qmail?	19
5.11	Dove è un riproduttore per Flash (SWF)?	20
5.12	Dove è Google Earth?	20
5.13	Dove è il software VoIP?	20
5.14	Ho una scheda di rete wireless che non funziona con Linux. Cosa fare?	20
6	Gli archivi FTP Debian	21
6.1	Quante distribuzioni di Debian ci sono?	21
6.2	Cosa sono tutti quei nomi come etch, lenny, ecc.?	21
6.2.1	Quali altri nomi in codice sono stati usati in passato?	21
6.2.2	Da dove derivano questi nomi in codice?	21
6.3	Cosa dire a proposito di "sid"?	22

6.4	Cosa contiene la <i>directory stable</i> ?	22
6.5	Cosa contiene la distribuzione <i>testing</i> ?	23
6.5.1	Cosa dire di “ <i>testing</i> ”? Come viene “congelata”?	23
6.6	Cosa contiene la distribuzione <i>unstable</i> ?	23
6.7	Cosa sono tutte quelle <i>directory</i> negli archivi FTP Debian?	23
6.8	Cosa sono tutte quelle <i>directory</i> dentro a <i>dists/stable/main</i> ?	24
6.9	Dove è il codice sorgente?	24
6.10	Cosa c'è nella <i>directory pool</i> ?	24
6.11	Cos'è “ <i>incoming</i> ”?	25
6.12	Come impostare un proprio repository usabile con <i>apt</i> ?	25
7	Fondamenti sul sistema di gestione dei pacchetti Debian	27
7.1	Cosa è un pacchetto Debian?	27
7.2	Qual è il formato di un pacchetto binario Debian?	28
7.3	Perché i nomi dei pacchetti Debian sono così lunghi?	28
7.4	Cos'è un file di controllo Debian?	28
7.5	Cos'è un <i>conf</i> file Debian?	29
7.6	Cosa sono gli script <i>preinst</i> , <i>postinst</i> , <i>prepm</i> e <i>postrm</i> di Debian?	29
7.7	Cosa sono i pacchetti <i>Essential</i> , <i>Required</i> , <i>Important</i> , <i>Standard</i> , <i>Optional</i> o <i>Extra</i> ?	30
7.8	Cos'è un pacchetto virtuale?	30
7.9	Cosa significa dire che un pacchetto <i>Dipende da</i> , <i>Raccomanda</i> , <i>Suggerisce</i> , <i>Va in conflitto con</i> , <i>Sostituisce</i> , <i>Rompe</i> o <i>Fornisce</i> un altro pacchetto?	31
7.10	Cosa significa <i>Pre-Depends</i> ?	31
7.11	Cosa significano <i>sconosciuto</i> , <i>installa</i> , <i>rimuovi</i> , <i>elimina</i> e <i>blocca</i> nello stato di un pacchetto?	31
7.12	Come si blocca un pacchetto?	32
7.13	Come si installa un pacchetto sorgente?	32
7.14	Come si compilano pacchetti binari da un pacchetto sorgente?	33
7.15	Come ci si creano pacchetti Debian da sé?	33
8	Gli strumenti di gestione dei pacchetti Debian	35
8.1	Quali programmi fornisce Debian per gestire i suoi pacchetti?	35
8.1.1	<i>dpkg</i>	35
8.1.2	<i>APT</i>	36
8.1.3	<i>aptitude</i>	37
8.1.4	<i>synaptic</i>	38
8.1.5	<i>tasksel</i>	38
8.1.6	Altri strumenti per la gestione dei pacchetti	38
8.2	Debian afferma di essere in grado di aggiornare un programma in esecuzione; come viene fatto?	38
8.3	Come si può sapere quali pacchetti sono già installati su un sistema Debian?	38
8.4	How do I display the files of an installed package?	39
8.5	Come posso scoprire quale pacchetto ha prodotto un particolare file?	39
8.6	Why is ‘foo-data’ not removed when I uninstall ‘foo’? How do I make sure old unused library-packages get purged?	39

9	Mantenere il sistema Debian aggiornato	41
9.1	Come si può mantenere il proprio sistema Debian aggiornato?	41
9.1.1	aptitude	41
9.1.2	apt-get e apt-cdrom	42
9.1.3	mirror	42
9.2	È necessario entrare in modalità singolo-utente per aggiornare un pacchetto?	42
9.3	È necessario tenere tutti questi file di archivio .deb sul disco?	43
9.4	How can I keep a log of the packages I added to the system? I'd like to know when upgrades and removals have occurred and on which packages!	43
9.5	È possibile aggiornare automaticamente il sistema?	43
9.6	Se si hanno svariate macchine come è possibile scaricare gli aggiornamenti una volta sola?	43
10	Debian e il kernel	45
10.1	Si può installare e compilare un kernel senza alcun adattamento specifico per Debian?	45
10.2	Quali strumenti fornisce Debian per costruire kernel personalizzati?	45
10.3	Quali strumenti speciali fornisce Debian per lavorare con i moduli?	45
10.4	Si può disinstallare in sicurezza un vecchio pacchetto kernel e, se sì, come?	45
10.5	Where can I get more information about Linux packages for Debian?	46
11	Personalizzare il proprio sistema Debian GNU/Linux	47
11.1	Come ci si può assicurare che tutti i programmi usino lo stesso formato per la carta?	47
11.2	Come si può fornire accesso alle periferiche hardware senza compromettere la sicurezza?	47
11.3	Come caricare un tipo di carattere per la console all'avvio nella maniera Debian?	47
11.4	Come si possono configurare le impostazioni predefinite per l'applicazione di un programma X11?	48
11.5	Ogni distribuzione sembra avere un metodo di avvio differente. Cosa dire di quello di Debian?	48
11.6	Quale altre funzionalità sono fornite per personalizzare il processo di avvio a parte <code>rc.local</code> ?	48
11.7	Come si comporta il sistema di manutenzione dei pacchetti nel caso di pacchetti che contengono file di configurazione per altri pacchetti?	49
11.8	Come sovrascrivere un file installato da un pacchetto in modo che ne venga usata una versione differente?	49
11.9	Come si può far sì che un pacchetto generato localmente venga incluso nella lista dei pacchetti disponibili che il sistema di gestione dei pacchetti conosce?	50
11.10	Ad alcuni utenti piace mawk, ad altri gawk; ad alcuni piace vim, ad altri elvis; ad alcuni piace trn, ad altri tin; come supporta Debian le diversità?	50
12	Ottenere supporto per Debian GNU/Linux	53
12.1	Quale altra documentazione esiste su e per un sistema Debian?	53
12.2	Ci sono risorse on-line dove discutere su Debian?	54
12.2.1	Mailing-list	54
12.2.2	Forum web	54
12.2.3	Wiki	54
12.2.4	Manutentori	55
12.2.5	Gruppi di discussione (newsgroup) Usenet	55
12.3	Esiste un modo rapido per cercare informazioni su Debian GNU/Linux?	55
12.4	Ci sono registri dei bug conosciuti?	55
12.5	Come si segnala un bug in Debian?	55

13 Contribuire al Progetto Debian	57
13.1 How can I become a Debian member/Debian developer?	57
13.2 Come si possono fornire risorse al progetto Debian?	57
13.3 Come si può contribuire finanziariamente al progetto Debian?	57
13.3.1 Software in the Public Interest	57
13.3.2 Altre organizzazioni	58
14 Ridistribuire Debian GNU/Linux in un prodotto commerciale	59
14.1 Si possono creare e vendere CD Debian?	59
14.2 Si può impacchettare Debian con software non libero?	59
14.3 Sto creando una speciale distribuzione Linux per un "mercato verticale". Posso usare Debian GNU/Linux come nucleo di un sistema Linux e aggiungere le mie applicazioni su di esso?	59
14.4 È possibile mettere un proprio programma commerciale in un "pacchetto" Debian cosicché si possa installare senza fatica su ogni sistema Debian?	60
15 Cambiamenti attesi nel prossimo rilascio principale di Debian	61
15.1 Rafforzare il sistema	61
15.2 Supporto esteso per gli utenti non-inglesi	61
15.3 Miglioramenti nell'installatore Debian	62
15.4 Più architetture	62
15.5 Più kernel	62
16 Informazioni generali su questa FAQ	63
16.1 Autori	63
16.2 Feedback	63
16.3 Reperibilità	63
16.4 Formato del documento	64

Capitolo 1

Definizioni e introduzione

1.1 Cosa sono queste FAQ?

Questo documento contiene domande frequenti (e le risposte!) sulla distribuzione Debian (Debian GNU/Linux e altre) e sul progetto Debian. Se possibile vengono forniti riferimenti ad altra documentazione: in questo testo non vengono citati grossi brani di documentazione esterna. Si potrà notare che alcune risposte danno per scontata una certa conoscenza dei sistemi operativi *nix. Si è cercato di dare per scontata una quantità di conoscenze più piccola possibile: le risposte a domande generiche dei principianti sono mantenute semplici.

Se non si trova ciò che si sta cercando in queste FAQ, assicurarsi di controllare 'Quale altra documentazione esiste su e per un sistema Debian?' a pagina 53. Se anche ciò non aiuta, fare riferimento a 'Feedback' a pagina 63.

1.2 Cosa è Debian GNU/Linux?

Debian GNU/Linux è una particolare *distribuzione* del sistema operativo Linux e di numerosi pacchetti funzionanti su di essa.

Debian GNU/Linux è:

- **completa:** Debian include attualmente più di 42551 pacchetti software. Gli utenti possono scegliere quali pacchetti installare; Debian fornisce uno strumento a questo scopo. Si può trovare un elenco con la descrizione dei pacchetti attualmente disponibili in Debian in uno qualsiasi dei siti mirror (<http://www.debian.org/distrib/ftplist>) di Debian.
- **liberamente utilizzabile e ridistribuibile:** non è necessaria l'affiliazione a nessuna associazione né alcun pagamento per partecipare alla sua distribuzione e al suo sviluppo. Tutti i pacchetti che sono formalmente parte di Debian GNU/Linux sono liberamente ridistribuibili, di solito nei termini specificati dalla licenza GNU GPL (GNU General Public License).
Gli archivi FTP Debian contengono anche approssimativamente 696 pacchetti software (nelle sezioni `non-free` e `contrib`), che sono distribuibili nei termini specifici indicati in ciascun pacchetto.
- **dinamica:** con circa 1033 volontari che contribuiscono costantemente con codice nuovo e migliorato, Debian è in rapida evoluzione. Gli archivi FTP vengono aggiornati due volte al giorno.

La maggior parte degli utenti Linux usa una particolare *distribuzione* di Linux, come Debian GNU/Linux. In via di principio, però, gli utenti possono ottenere il kernel Linux da Internet o da qualche altra parte e compilarlo loro stessi. Poi, nello stesso modo, possono ottenere il codice sorgente di molte applicazioni, compilare i programmi e installarli nel proprio sistema. Per programmi complessi questo processo può essere non solo un dispendio di tempo, ma può anche comportare degli errori. Per evitare tutto ciò, gli utenti spesso scelgono di ottenere il sistema operativo ed i pacchetti delle applicazioni da uno dei distributori Linux. Quello che distingue i vari distributori Linux è il software, i protocolli ed i procedimenti usati per impacchettare, installare e tenere traccia dei pacchetti applicativi sui sistemi degli utenti, e inoltre gli strumenti di installazione e manutenzione, documentazione ed altri servizi.

Debian GNU/Linux è il risultato di uno sforzo di volontari per creare un sistema operativo compatibile con Unix, di alta qualità, libero, completato da un insieme di applicazioni. L'idea di un sistema operativo *nix libero ha avuto origine dal progetto GNU e molte applicazioni che rendono Debian GNU/Linux così utile sono state sviluppate dal progetto GNU.

Per Debian, libero ha il significato del mondo GNU (si vedano le Linee Guida Debian per il Software Libero (http://www.debian.org/social_contract#guidelines)). Quando parliamo di software libero, ci riferiamo alla libertà, non al prezzo. Software libero significa che si ha la libertà di distribuire copie di software libero, che si riceve il codice sorgente o lo si può ottenere se lo si desidera, che si può cambiare il software o usare parte di esso in nuovi programmi liberi e che si è a conoscenza di poter fare queste cose.

Il Progetto Debian è stato creato da Ian Murdock nel 1993, inizialmente sotto la sponsorizzazione del progetto GNU della Free Software Foundation. Oggi, gli sviluppatori Debian lo vedono come un diretto discendente del progetto GNU.

Tuttavia anche se Debian GNU/Linux in sé è software libero, rappresenta una base su cui possono essere costruite distribuzioni di Linux dal valore aggiunto. Fornendo un sistema di base sicuro e completo di funzionalità, Debian fornisce agli utenti Linux una compatibilità crescente e permette ai creatori di distribuzioni Linux di eliminare sforzi duplicati e di concentrarsi sulle caratteristiche che rendono speciale la propria distribuzione. Vedere ‘Sto creando una speciale distribuzione Linux per un “mercato verticale”’. Posso usare Debian GNU/Linux come nucleo di un sistema Linux e aggiungere le mie applicazioni su di esso?’ a pagina 59 per maggiori informazioni.

1.3 OK, ora so cosa è Debian... cosa è Linux?!

In breve, Linux è il kernel di un sistema operativo *nix. È stato originariamente progettato per PC 386 (e superiori); ora può girare anche su dozzine di altri sistemi. Linux è stato scritto da Linus Torvalds e molti altri informatici di tutto il mondo.

Oltre al suo kernel, un sistema “Linux” di solito ha:

- un file system che segue il Linux Filesystem Hierarchy Standard <http://www.pathname.com/fhs/>;
- una vasta gamma di utilità Unix, molte delle quali sono state sviluppate dal progetto GNU e dalla Free Software Foundation.

La combinazione del kernel Linux, del file system, delle utilità GNU, di quelle FSF e delle altre utilità è stata progettata per raggiungere la compatibilità con lo standard POSIX (IEEE 1003.1); si veda ‘Quanto è compatibile Debian con altri sistemi Unix a livello di codice sorgente?’ a pagina 14.

Per maggiori informazioni su Linux, vedere Cosa è Linux (<http://www.linux.org/info/>) di Linux Online (<http://www.linux.org/>).

1.4 Debian fa solo GNU/Linux?

Attualmente, Debian è disponibile solo per Linux, ma con Debian GNU/Hurd e Debian su kernel BSD si è iniziato ad offrire come piattaforme di sviluppo, server e desktop anche altri sistemi operativi non basati su Linux. Questi port non-Linux tuttavia non sono ancora rilasciati ufficialmente.

Il primo port su cui si è lavorato è Debian GNU/Hurd.

Hurd è un insieme di servizi che funzionano sul microkernel Mach GNU. Insieme costituiscono la base del sistema operativo GNU.

Si veda <http://www.gnu.org/software/hurd/> per maggiori informazioni su GNU/Hurd in generale e <http://www.debian.org/ports/hurd/> per maggiori informazioni su Debian GNU/Hurd.

A second effort is the port to a BSD kernel. People are working with the FreeBSD kernel.

Si veda <http://www.debian.org/ports/#nonlinux> per maggiori informazioni su questi port non-Linux.

1.5 Qual è la differenza tra Debian GNU/Linux e le altre distribuzioni Linux? Perché scegliere Debian invece di qualche altra distribuzione?

Queste caratteristiche chiave distinguono Debian dalle altre distribuzioni Linux:

Libertà: Come indicato nel Contratto sociale Debian (http://www.debian.org/social_contract), Debian rimarrà libera al 100%. Debian è molto rigorosa per ciò che riguarda la distribuzione di software veramente libero. Le linee guida usate per determinare se un lavoro è “libero” sono fornite nelle Linee guida Debian per il Software Libero (DFSG) (http://www.debian.org/social_contract#guidelines).

Il sistema di manutenzione dei pacchetti Debian: L'intero sistema, o qualsiasi sua componente individuale, può essere aggiornato in situ senza riformattare, senza perdere i file di configurazione personalizzati e, nella maggior parte dei casi, senza riavviare il sistema. La maggior parte delle distribuzioni Linux disponibili attualmente ha qualche tipo di sistema di manutenzione dei pacchetti; il sistema di manutenzione dei pacchetti Debian è unico e particolarmente robusto (si veda 'Fondamenti sul sistema di gestione dei pacchetti Debian' a pagina 27).

Sviluppo aperto: Mentre molte altre distribuzioni Linux sono sviluppate da singoli, gruppi piccoli, chiusi o produttori commerciali, Debian è un'importante distribuzione Linux che è sviluppata da una comunità di individui che hanno fatta propria la causa della creazione di un sistema operativo libero, nello stesso spirito di Linux e di altro software libero.

Più di 1033 volontari che mantengono i pacchetti stanno lavorando su più di 42551 pacchetti migliorando Debian GNU/Linux. Gli sviluppatori Debian contribuiscono al progetto non scrivendo nuove applicazioni (nella maggior parte dei casi), ma impacchettando software già esistente in accordo con gli standard del progetto, comunicando segnalazioni sui bug agli sviluppatori originali e fornendo supporto agli utenti. Si vedano anche le informazioni aggiuntive su come diventare un contributore in 'How can I become a Debian member/Debian developer?' a pagina 57.

Il Sistema Operativo Universale: Debian viene fornita con più di 42551 pacchetti (<http://packages.debian.org/stable/>) e gira su 10 architetture (<http://www.debian.org/ports/>). Ciò è molto più di quello che è disponibile per le altre distribuzioni GNU/Linux. Si veda 'Quali tipi di applicazioni e software di sviluppo sono disponibili per Debian GNU/Linux?' a pagina 17 per una panoramica del software fornito e 'Su quali architetture o sistemi hardware funziona Debian?' a pagina 13 per una descrizione delle piattaforme hardware supportate.

Il Sistema di tracciamento dei bug: La dispersione geografica degli sviluppatori Debian ha richiesto degli strumenti sofisticati ed una comunicazione rapida dei bug e delle soluzioni per accelerare lo sviluppo del sistema. Gli utenti sono incoraggiati a segnalare i bug in uno stile formale, che sia rapidamente accessibile attraverso gli archivi WWW o via posta elettronica. Si vedano le informazioni aggiuntive in questa FAQ sulla gestione dei log dei bug in 'Ci sono registri dei bug conosciuti?' a pagina 55.

La Debian Policy: Debian ha un'esauriente specifica dei propri standard di qualità, la Debian Policy. Questo documento definisce la qualità e gli standard che sono richiesti ai pacchetti Debian.

Per ulteriori informazioni su questo, si veda la pagina web sulle ragioni per scegliere Debian (http://www.debian.org/intro/why_debian).

1.6 Come si rapporta il progetto Debian con il progetto GNU della Free Software Foundation?

Il sistema Debian è costruito sugli ideali del software libero fortemente sostenuto in origine dalla Free Software Foundation (<http://www.gnu.org/>) ed in particolare da Richard Stallman (<http://www.stallman.org/>). I potenti strumenti, utilità e applicazioni per lo sviluppo del sistema della FSF rappresentano anche una parte chiave del sistema Debian.

Il Progetto Debian è un'entità separata dalla FSF, comunque i due comunicano regolarmente e cooperano su vari progetti. La FSF ha richiesto esplicitamente che il nostro sistema venisse chiamato "Debian GNU/Linux" e siamo felici di soddisfare questa richiesta.

L'obiettivo di sempre della FSF è stato quello di sviluppare un nuovo sistema operativo chiamato GNU basato su Hurd (<http://www.gnu.org/software/hurd/>). Debian sta lavorando con la FSF su questo sistema, chiamato Debian GNU/Hurd (<http://www.debian.org/ports/hurd/>).

1.7 Come si pronuncia Debian e cosa significa questa parola?

Il nome del progetto si pronuncia Deb'-i-en, con una e breve in Deb e con l'accento sulla prima sillaba. Questa parola è la contrazione dei nomi di Debra e Ian Murdock, che fondarono il progetto. (I dizionari sembrano essere ambigui sulla pronuncia di Ian (!), ma Ian preferisce i'-en.)

Capitolo 2

Ottenere e installare Debian GNU/Linux

La documentazione ufficiale con le istruzioni per l'installazione è la Guida all'installazione di Debian GNU/Linux (<http://www.debian.org/releases/stable/installmanual>). Qui vengono fornite alcune note aggiuntive su come ottenere ed installare Debian GNU/Linux.

2.1 Qual è la versione più recente di Debian?

Attualmente ci sono tre versioni di Debian GNU/Linux:

il rilascio 8, alias la distribuzione "stabile" o jessie Questo è software stabile e ben testato, cambia se vengono incorporate soluzioni importanti a problemi di sicurezza e usabilità.

la distribuzione "testing", attualmente chiamata stretch Qui è dove vengono messi i pacchetti che saranno rilasciati come prossima "stable"; sono stati testati un po' in unstable ma possono non essere ancora completamente adatti ad essere rilasciati. Questa distribuzione è aggiornata più frequentemente della "stable", ma non della "unstable".

la distribuzione "unstable" Questa è la versione attualmente in fase di sviluppo; è aggiornata continuamente. Si possono recuperare i pacchetti dall'archivio "unstable" su ogni sito FTP Debian e usarli per aggiornare il proprio sistema in ogni momento, ma non bisogna aspettarsi che il sistema sia utilizzabile o stabile come prima: ecco perché è chiamata "unstable (instabile)"!

Si veda 'Quante distribuzioni di Debian ci sono?' a pagina 21 per maggiori informazioni.

2.2 Esistono aggiornamenti dei pacchetti in "stable"?

Nessuna nuova funzionalità viene aggiunta al rilascio stabile. Una volta che una versione di Debian viene rilasciata ed etichettata come "stable", avrà solo gli aggiornamenti di sicurezza. Cioè saranno aggiornati solo i pacchetti in cui è stata trovata dopo il rilascio una vulnerabilità in termini di sicurezza. Tutti gli aggiornamenti di sicurezza vengono forniti attraverso security.debian.org (<ftp://security.debian.org>).

Gli aggiornamenti di sicurezza servono ad un solo scopo: fornire soluzioni per vulnerabilità nella sicurezza. Non sono un metodo per infiltrare cambiamenti ulteriori nel rilascio stabile senza passare dalla normale procedura dei rilasci minori. Di conseguenza, le soluzioni per i pacchetti con problemi di sicurezza non aggiornano il software. Il Team per la sicurezza di Debian farà invece il backport delle soluzioni necessarie nella versione del software distribuita in "stable".

Per maggiori informazioni sul supporto per la sicurezza si legga la Security FAQ (<http://www.debian.org/security/faq>) o il Debian Security Manual (<http://www.debian.org/doc/manuals/securing-debian-howto/>).

2.3 Dove/Come si possono ottenere le immagini di installazione di Debian?

Le immagini per l'installazione si possono ottenere scaricando i file appropriati da uno dei siti web Debian (<http://www.debian.org/mirror/list>).

Per maggiori informazioni su immagini di CD (e DVD) si faccia riferimento a Debian GNU/Linux su CD (<http://www.debian.org/CD>).

2.4 Come si installa Debian da CD-ROM?

Installare Debian da CD è semplicissimo: configurare il proprio sistema in modo che si avvii da CD, inserire il CD e riavviare. Il sistema avrà allora in esecuzione l'installatore Debian. Per maggiori informazioni si veda la Guida all'installazione di Debian GNU/Linux (<http://www.debian.org/releases/stable/installmanual>).

2.5 Perché il CD-ROM ufficiale del rilascio "stable" contiene collegamenti simbolici "frozen" e "unstable"? Questo CD doveva contenere solo "stable"!

Le immagini dei CD ufficiali di Debian contengono effettivamente collegamenti simbolici quali:

```
/dists/frozen -> jessie/  
/dists/stable -> jessie/  
/dists/testing -> jessie/  
/dists/unstable -> jessie/
```

in modo da poter funzionare quando il file `sources.list` ha voci come

```
deb cdrom:[<nome come sulla etichetta del CD>] /unstable main [...]
```

Il fatto che siano presenti questi collegamenti simbolici *non* significa che l'immagine è di "unstable" o "testing" o altro. Si legga l'etichetta del CD in `/.disk/info` per scoprire quale versione di Debian esso contenga. Questa informazione è presente anche nel file `/README.txt` sul CD.

Si legga <http://www.debian.org/releases/> per scoprire quali siano gli attuali rilasci "stable" e "testing".

2.6 È possibile ottenere e installare Debian direttamente da un sito Internet remoto?

Sì. Si può avviare il sistema di installazione Debian da un insieme di file scaricabili dal nostro sito FTP e dai suoi mirror.

Si può scaricare una piccola immagine di CD, creare da essa un CD avviabile, installare il sistema di base con esso ed il resto dalla rete. Per maggiori informazioni si veda <http://www.debian.org/CD/netinst/>.

2.7 Esistono strategie alternative per avviare l'installatore di sistema?

Sì. Oltre che da CD o DVD, si può installare Debian GNU/Linux avviando da chiavette USB, direttamente da hard disk o usando l'avvio di rete TFTP. Per l'installazione su più computer è possibile fare installazioni completamente automatiche. N.B.: non tutti i metodi sono supportati da tutte le architetture. Una volta avviato l'installatore, il resto del sistema può essere scaricato dalla rete o installato da supporti locali. Per maggiori informazioni si veda la Guida all'installazione di Debian GNU/Linux (<http://www.debian.org/releases/stable/installmanual>).

Capitolo 3

Scegliere una distribuzione Debian

Ci sono molte diverse distribuzioni Debian. Scegliere la distribuzione Debian adatta è una decisione importante. Questa sezione fornisce alcune informazioni utili per gli utenti che desiderano fare la scelta più adatta per il loro sistema e inoltre risponde ad alcune possibili domande che potrebbero nascere al momento della scelta. Non parla del “perché si dovrebbe scegliere Debian” ma piuttosto di “quale distribuzione di Debian scegliere”.

Per maggiori informazioni sulle distribuzioni disponibili si veda ‘Quante distribuzioni di Debian ci sono?’ a pagina [21](#).

3.1 Quale distribuzione Debian (stable/testing/unstable) è meglio per me?

La risposta è piuttosto complessa. Dipende veramente da cosa si ha in mente di fare. Una soluzione potrebbe essere chiedere ad un amico che usa Debian. Ma ciò non significa che non è possibile prendere una decisione in modo autonomo. Di fatto, si dovrebbe essere in grado di fare una scelta una volta completata la lettura di questo capitolo.

- Se la sicurezza o la stabilità sono aspetti di una qualche importanza: installare stable. Punto. Questa è la scelta consigliata.
- Se si è un nuovo utente che deve installare su una macchina desktop, iniziare con stable. Parte del software è piuttosto vecchio, ma è l’ambiente di lavoro con meno bug. Si potrà facilmente passare alla più moderna unstable (o testing) una volta che si è più sicuri di sé.
- If you are a desktop user with a lot of experience in the operating system and do not mind facing the odd bug now and then, or even full system breakage, use unstable. It has all the latest and greatest software, and bugs are usually fixed swiftly.
- Se si gestisce un server, specialmente uno per cui la stabilità è un requisito importante o che è esposto ad Internet, installare stable. Questa è di gran lunga la scelta più sicura e affidabile.

Le domande che seguono forniscono, si spera, ulteriori dettagli su queste scelte. Dopo aver letto tutte queste FAQ, se ancora non si è in grado di prendere una decisione, optare per la distribuzione stable.

3.1.1 È stato suggerito di installare stable, ma in essa l’hardware pincopallino non viene rilevato o non funziona. Cosa fare?

Provare a cercare nel web usando un motore di ricerca e vedere se qualcun altro è stato in grado di farlo funzionare in stable. La maggior parte dell’hardware dovrebbe funzionare senza problemi in stable. Se però si ha dell’hardware recentissimo e all’avanguardia, questo potrebbe non funzionare con stable. Se questa è la situazione, si può installare testing o unstable o aggiornare il sistema ad una di esse.

Per i portatili, <http://www.linux-on-laptops.com/> è un sito web molto buono per vedere se qualcun altro è stato in grado di farli funzionare con Linux. Il sito non è specifico per Debian, ma ciò nonostante è una risorsa preziosissima. Non conosco un sito web simile per i desktop.

Another option would be to ask in the debian-user mailing list by sending an email to debian-user@lists.debian.org. Messages can be posted to the list even without subscribing. The archives can be read through <http://lists.debian.org/debian-user/>. Information regarding subscribing to the list can be found at the location of archives. You are strongly encouraged to post your questions on the mailing-list rather than on irc (<http://www.debian.org/support>). The mailing-list messages are archived, so the solution to your problem can help others with the same issue.

3.1.2 Nelle differenti distribuzioni ci sono versioni diverse dei pacchetti?

Sì. Unstable ha le versioni più recenti, ma i suoi pacchetti non sono ben testati e potrebbero avere dei bug.

D'altro canto, stable contiene versioni vecchie dei pacchetti, ma esse sono ben testate ed è meno probabile che abbiano un qualche bug.

I pacchetti in testing sono una via di mezzo tra questi due estremi.

3.1.3 Le distribuzioni stable contengono pacchetti veramente datati. Basta guardare Kde, Gnome, Xorg o persino il kernel: sono molto vecchi. Perché?

Beh, questo può essere vero. L'età dei pacchetti in stable dipende da quando è stato fatto l'ultimo rilascio. Dato che di solito trascorre più di 1 anno da un rilascio all'altro, stable potrebbe contenere versioni vecchie dei pacchetti. Tuttavia sono state testate per diritto e per rovescio. Si può dire a ragion veduta che i pacchetti non hanno alcun bug importante noto, falle di sicurezza, ecc. I pacchetti in stable si integrano perfettamente gli uni con gli altri. Queste caratteristiche sono molto importanti per server di produzione che devono essere in funzione 24 ore al giorno, 7 giorni alla settimana.

On the other hand, packages in testing or unstable can have hidden bugs, security holes etc. Moreover, some packages in testing and unstable might not be working as intended. Usually people working on a single desktop prefer having the latest and most modern set of packages. Unstable is the solution for this group of people.

Come si può vedere, la stabilità e l'estrema modernità sono ai capi opposti dello spettro. Se è necessaria la stabilità, installare la distribuzione stable. Se si vuole lavorare con i pacchetti più recenti, allora installare unstable.

3.1.4 Se si decidesse di passare ad un'altra distribuzione, sarebbe possibile farlo?

Sì, ma è un processo unidirezionale. Si può fare il passaggio stable → testing → unstable. La direzione inversa invece non è "possibile". Perciò è bene essere sicuri se si ha intenzione di installare unstable o di aggiornare il sistema ad essa.

Actually, if you are an expert and if you are willing to spend some time and if you are real careful and if you know what you are doing, then it might be possible to go from unstable to testing and then to stable. The installer scripts are not designed to do that. So in the process, your configuration files might be lost and...

3.1.5 Potete dirmi se installare stable, testing o unstable?

No. This is a rather subjective issue. There is no perfect answer as it depends on your software needs, your willingness to deal with possible breakage, and your experience in system administration. Here are some tips:

- Stable è solida come una roccia. Non diventa difettosa e ha il pieno supporto di sicurezza. Ma potrebbe non avere il supporto per l'hardware più recente.
- Testing ha software più aggiornato di Stable e diventa difettosa meno di frequente di Unstable, ma quando lo fa potrebbe volerci parecchio tempo prima che le cose vengano aggiustate. A volte possono volerci giorni e a volte mesi. Inoltre non ha un supporto di sicurezza permanente.
- Unstable ha il software più recente e cambia molto. Di conseguenza può danneggiarsi in qualunque momento. Tuttavia, le cose vengono risolte spesso in un paio di giorni ed ha sempre le ultime versioni dei pacchetti software per Debian.

Quando si deve scegliere tra testing e unstable tenere in considerazione che di potrebbero essere casi in cui sarebbe meglio seguire testing invece di unstable. Uno degli autori di questo documento si è trovato in una di queste situazioni a causa della transizione di gcc da gcc3 a gcc4. Stava cercando di installare il pacchetto `labplot` su una macchina con unstable e l'installazione non era lì possibile perché alcune delle dipendenze avevano già fatto il passaggio a gcc4 ed alcune no. Il pacchetto in testing però era installabile su una macchina testing dato che i pacchetti che avevano fatto la transizione a gcc4 non avevano ancora raggiunto testing.

3.1.6 È stato detto che testing si può rompere. Cosa significa?

A volte, un pacchetto può non essere installabile tramite gli strumenti di gestione dei pacchetti. Altre volte, un pacchetto può proprio non essere disponibile: forse è stato (temporaneamente) rimosso a causa di bug o di dipendenze non soddisfatte. Altre volte ancora, un pacchetto si installa ma non si comporta nel modo corretto.

Quando ciò avviene, si dice che la distribuzione è rotta (almeno per quel che riguarda il pacchetto in questione).

3.1.7 Come mai testing può essere danneggiata per mesi? Le soluzioni introdotte in unstable non passano direttamente in testing?

The bug fixes and improvements introduced in the unstable distribution trickle down to testing after a certain number of days. Let's say this threshold is 5 days. The packages in unstable go into testing only when there are no RC-bugs reported against them. If there is a RC-bug filed against a package in unstable, it will not go into testing after the 5 days.

The idea is that, if the package has any problems, it would be discovered by people using unstable and will be fixed before it enters testing. This keeps testing in a usable state for most of the time. Overall a brilliant concept, if you ask me. But things aren't always that simple. Consider the following situation:

- Immaginiamo di essere interessati al pacchetto XYZ.
- Ipotizziamo che al 10 giugno la versione in testing sia XYZ-3.6 e quella in unstable sia XYZ-3.7.
- After 5 days, XYZ-3.7 from unstable migrates into testing.
- So on June 15, both testing and unstable have XYZ-3.7 in their repositories.
- Diciamo che un utente della distribuzione testing vede che è disponibile un nuovo pacchetto XYZ e aggiorna la propria versione XYZ-3.6 a XYZ-3.7.
- Ora diciamo che il 25 giugno qualche utente di testing o unstable scopre che c'è un bug critico per il rilascio in XYZ-3.7 e lo segnala nel BTS.
- Il manutentore di XYZ corregge il bug e carica la versione corretta in unstable, diciamo il 30 giugno. Qui è stato ipotizzato che ci vogliano 5 giorni prima che il manutentore corregga il bug e carichi la nuova versione. Il numero 5 non deve essere preso letteralmente: potrebbe essere di più o di meno a seconda del grado di severità del baco critico in questione.
- This new version in unstable, XYZ-3.8 is scheduled to enter testing on July 5th.
- But on July 3rd some other person discovers another RC-bug in XYZ-3.8.
- Diciamo che il manutentore di XYZ risolve questo nuovo bug critico e carica la nuova versione di XYZ dop 5 giorni.
- So on July 8th, testing has XYZ-3.7 while unstable has XYZ-3.9.
- This new version XYZ-3.9 is now rescheduled to enter testing on July 13th.
- Now since you are running testing, and since XYZ-3.7 is buggy, you could probably use XYZ only after July 13th. That is you essentially ended up with a broken XYZ for about one month.

The situation can get much more complicated, if say, XYZ depends on 4 other packages. This could in turn lead to an unusable testing distribution for months. While the scenario above is imaginary, similar things can occur in real life, though they are rare.

3.1.8 Dal punto di vista di un amministratore, quale distribuzione richiede più attenzioni?

One of the main reasons why many people choose Debian over other Linux distributions is that it requires very little administration. People want a system that just works. In general one can say that stable requires very little maintenance, while testing and unstable require constant maintenance from the administrator. If you are running stable, all you need to worry about is keeping track of security updates. If you are running either testing or unstable it is a good idea to be aware of the new bugs discovered in the installed packages, new bugfixes/features introduced etc.

3.1.9 Cosa succede quando viene fatto un nuovo rilascio?

Questa domanda non aiuta a scegliere una distribuzione Debian, ma prima o poi verrà fatto di porsela.

The stable distribution is currently jessie; The next stable distribution will be called stretch. Let's consider the particular case of what happens when stretch is released as the new stable version.

- oldstable = wheezy; stable = jessie; testing = stretch; unstable = sid
- Ci si riferisce ad unstable sempre come a sid indipendentemente dal fatto che venga fatto o meno un rilascio.

- I pacchetti migrano continuamente da sid a testing (cioè stretch). I pacchetti in stable (cioè jessie) però rimangono gli stessi tranne che per gli aggiornamenti di sicurezza.
- Dopo un certo periodo testing viene congelata nello stato di freeze; ma continuerà ad essere chiamata testing. A questo punto nessun pacchetto nuovo può migrare da unstable a testing a meno che non contenga la soluzione ad un bug critico per il rilascio (RC).
- When testing is frozen, all the new bugfixes introduced have to be manually checked by the members of the release team. This is done to ensure that there won't be any unknown severe problems in the frozen testing.
- I bug RC nella "testing congelata" sono ridotti a zero oppure, se maggiori di zero, i bug sono contrassegnati come ignorati per il rilascio o sono posposti ad un rilascio minore.
- The 'frozen testing' with no rc-bugs will be released as the new stable version. In our example, this new stable release will be called stretch.
- A questo punto si avrà oldstable = jessie, stable = stretch. Il contenuto di stable e della "testing congelata" è a questo punto lo stesso.
- Una nuova testing viene basata sulla vecchia testing.
- I pacchetti iniziano ad arrivare da sid in testing e la comunità Debian lavora ora per fare il prossimo rilascio stable.

3.1.10 Ho un desktop/cluster in funzione con installata Debian. Come fare a sapere quale distribuzione è in esecuzione?

Nella maggior parte dei casi scoprirlo è molto semplice. Guardare il file `/etc/apt/sources.list`. Ci sarà una voce simile alla seguente:

```
deb http://ftp.us.debian.org/debian/ unstable main contrib
```

Il terzo campo ("unstable" nell'esempio precedente) indica la distribuzione Debian a cui il sistema sta attualmente facendo riferimento.

Si può anche usare `lsb_release` (disponibile nel pacchetto `lsb-release`). Se si esegue tale programma in un sistema unstable si ottiene:

```
$ lsb_release -a
LSB Version:    core-2.0-noarch:core-3.0-noarch:core-3.1-noarch:core-2.0-ia32:core-3.0-ia32:core-3.1-ia32
Distributor ID: Debian
Description:    Debian GNU/Linux unstable (sid)
Release:        unstable
Codename:       sid
```

Tuttavia non è sempre così facile. Alcuni sistemi possono avere file `sources.list` con voci multiple che corrispondono a distribuzioni diverse. Ciò può accadere se l'amministratore tiene traccia di pacchetti diversi da distribuzioni Debian diverse. Questa azione viene spesso chiamata `apt-pinning`. Questi sistemi possono avere in esecuzione un mix di distribuzioni.

3.1.11 I am currently tracking stable. Can I change to testing or unstable? If so, how?

Se si sta attualmente usando stable, allora nel file `/etc/apt/sources.list` il terzo campo sarà o «jessie» o «stable». È necessario cambiarlo e mettere la distribuzione che si vuole usare. Se si vuole usare testing, allora modificare il terzo campo di `/etc/apt/sources.list` in «testing». Se si vuole usare unstable, allora modificare il terzo campo in «unstable».

Currently testing is called stretch. So, if you change the third field of `/etc/apt/sources.list` to 'stretch', then also you will be running testing. But even when stretch becomes stable, you will still be tracking stretch.

Unstable si chiama sempre Sid. Perciò se si cambia il terzo campo di `/etc/apt/sources.list` in «sid», si userà unstable.

Attualmente Debian offre aggiornamenti di sicurezza per testing ma non per unstable, dato che le risoluzioni dei problemi in unstable vengono direttamente fatte all'archivio principale. Perciò se si usa unstable assicurarsi di rimuovere dal file `/etc/apt/sources.list` le righe relative agli aggiornamenti di sicurezza .

Se è disponibile un documento con le note di rilascio per la distribuzione a cui si sta per fare l'aggiornamento (anche se non ne è ancora stato fatto il rilascio) sarebbe saggio leggerlo, dato che potrebbe fornire informazioni su come fare l'aggiornamento.

Ciò nonostante, una volta fatti i cambiamenti detti sopra si può eseguire `aptitudeupdate` e poi installare i pacchetti desiderati. Notare che l'installazione di un pacchetto da una distribuzione diversa può aggiornare automaticamente metà del sistema. Se si installano pacchetti singoli si finirà per avere in esecuzione un sistema con un mix di distribuzioni.

Potrebbe essere meglio in alcune situazioni aggiornare semplicemente l'intero sistema alla nuova distribuzione eseguendo `apt full-upgrade`, `aptitude safe-upgrade` o `aptitude full-upgrade`. Per maggiori informazioni, leggere le pagine di manuale di `apt` e `aptitude`.

3.1.12 Attualmente sto usando testing (stretch). Cosa succederà quando verrà fatto un rilascio? Continuerò ad usare testing o la macchina userà la nuova distribuzione stabile?

Dipende dalle voci nel file `/etc/apt/sources.list`. Se si sta attualmente usando testing, tali voci saranno simili a una di queste due:

```
deb http://ftp.us.debian.org/debian/ testing main
```

o

```
deb http://ftp.us.debian.org/debian/ stretch main
```

Se nel terzo campo in `/etc/apt/sources.list` c'è "testing" allora si continuerà ad usare testing anche dopo un avvenuto rilascio. Perciò dopo che stretch sarà rilasciata, si starà usando una nuova distribuzione Debian che avrà un nome in codice diverso. I cambiamenti potrebbero non essere evidenti all'inizio, ma lo diverranno non appena nuovi pacchetti passeranno dalla distribuzione unstable a testing.

Se però il terzo campo contiene "stretch" allora si userà stable (dato che stretch sarà allora la nuova distribuzione stabile).

3.1.13 Sono ancora confuso. Cosa hai detto che devo installare?

If unsure, the best bet would be the stable distribution.

3.2 But what about Knoppix, Linux Mint Debian Edition, Ubuntu, and others?

Non sono Debian; sono *basate su Debian*. Benché ci siano molte somiglianze e cose in comune tra di esse, vi sono anche differenze fondamentali.

Tutte queste distribuzioni hanno i propri meriti e sono adatte a particolari tipi di utenti. Per maggiori informazioni, consultare la pagina sulle distribuzioni software basate su Debian (<http://www.debian.org/misc/children-distros>) disponibile sul sito web di Debian.

3.2.1 I know that Knoppix/Linux Mint Debian Edition/Ubuntu/... is Debian-based. So after installing it on the hard disk, can I use 'apt' package tools on it?

Queste distribuzioni sono basate su Debian. Ma non sono Debian. Sarà comunque possibile usare gli strumenti `apt` per i pacchetti facendo puntare il file `/etc/apt/sources.list` agli archivi di queste distribuzioni. Ma non si starà in quel caso usando Debian, si starà usando una distribuzione diversa. Non sono la stessa cosa.

Nella maggior parte delle situazioni, se si usa una distribuzione si dovrebbe rimanere legati ad essa e non mescolare pacchetti da altre distribuzioni. Molti problemi di funzionamento comuni derivano dall'uso di una distribuzione insieme al tentativo di installare pacchetti Debian da altre distribuzioni. Il fatto che usino lo stesso tipo di formato e lo stesso nome (`.deb`) non significa che siano automaticamente compatibili.

For example, Knoppix is a Linux distribution designed to be booted as a live CD whereas Debian is designed to be installed on the hard-disk. Knoppix is great if you want to know whether a particular piece of hardware works, or if you want to experience how a GNU/Linux system 'feels' etc., Knoppix is good for demonstration purposes while Debian is designed to run 24/7. Moreover the number of packages available, the number of architectures supported by Debian are far more than that of Knoppix.

Se si desidera Debian, la cosa migliore è installare Debian sin dall'inizio. Anche se è possibile installare Debian passando per altre distribuzioni, come Knoppix, la procedura richiede esperienza. Se si sta leggendo questa FAQ è lecito ipotizzare che si è alle prime armi sia con Debian sia con Knoppix. In questo caso, ci si risparmierà un sacco di problemi futuri se si installerà subito Debian.

3.2.2 I installed Knoppix/Linux Mint Debian Edition/Ubuntu/... on my hard disk. Now I have a problem. What should I do?

You are advised not to use the Debian forums (either mailing lists or IRC) for help as people there may base their suggestions on the assumption that you are running a Debian system. These “fixes” might not be suited to what you are running, and might even make your problem worse.

Usare prima i forum della distribuzione specifica che si sta usando. Se non si ottiene aiuto o l'aiuto ottenuto non risolve il problema, si può provare a chiedere nei forum Debian, ma tenere a mente l'avvertimento dato nel paragrafo precedente.

3.2.3 I'm using Knoppix/LMDE/Ubuntu/... and now I want to use Debian. How do I migrate?

Considerare il passaggio da una distribuzione basata su Debian a Debian come qualsiasi altro passaggio da un sistema operativo ad un altro. Si dovrebbe fare il backup di tutti i dati e reinstallare il sistema operativo da zero. Non si deve cercare di “aggiornare” a Debian usando gli strumenti di gestione dei pacchetti dato che ci si potrebbe ritrovare con un sistema non funzionante.

Se tutti i propri dati utente (cioè la propria `/home`) sono in una partizione separata la migrazione a Debian è in realtà piuttosto semplice, basta semplicemente dire al sistema di installazione di montare (ma non riformattare) quella partizione al momento della reinstallazione. È sempre caldamente consigliato fare il backup di tutti i propri dati oltre che della configurazione del sistema precedente (cioè `/etc/` e, forse, `/var/`).

Capitolo 4

Problemi di compatibilità

4.1 Su quali architetture o sistemi hardware funziona Debian?

Debian GNU/Linux include il codice sorgente completo per tutti i programmi inclusi, così dovrebbe funzionare su tutti i sistemi che sono supportati dal kernel Linux; si veda la FAQ Linux (<http://en.tldp.org/FAQ/Linux-FAQ/intro.html#DOES-LINUX-RUN-ON-MY-COMPUTER>) per i dettagli.

L'attuale rilascio 8 di Debian GNU/Linux contiene una distribuzione binaria completa per le seguenti architetture:

- *amd64*: questa copre i sistemi basati su CPU AMD a 64bit con estensione AMD64 e tutte le CPU Intel con estensione EM64T ed uno spazio utente a 64bit comune.
- *arm64*: per i dispositivi basati su ARM a 64 bit più recenti.
- *armel*: macchine ARM little-endian.
- *armhf*: un'alternativa ad *armel* per macchine ARMv7 con hard-float.
- *i386*: questa copre i sistemi basati su processori Intel e compatibili, inclusi gli Intel 386, 486, Pentium, Pentium Pro, Pentium II (sia Klamath che Celeron) e Pentium III e la maggior parte dei processori compatibili di AMD, Cyrix ed altri.
- *ia64*: computer Intel IA-64 ("Itanium").
- *mips*: sistemi MIPS big-endian della SGI, Indy e Indigo2; *mipsel*: macchine MIPS little-endian, DECstation Digital.
- *powerpc*: questa copre alcune macchine PowerPC IBM/Motorola, incluse i modelli PowerMac di Apple Macintosh e le macchine CHRP e PReP a architettura aperta.
- *ppc64el*: port per PowerPC a 64 bit little-endian, per diversi processori PowerPC/POWER recenti.
- *s390x*: port a 64 bit per macchine IBM System z, ha sostituito *s390*.

The development of binary distributions of Debian for *hurd-i386* (for GNU Hurd kernel on i386 32-bit PCs), *mipsel64* (for 64 bit MIPS in little-endian mode), *powerpcspe* (port for the "Signal Processing Engine" hardware), *sparc64* (for 64 bit SPARC processors), *sh* (for Hitachi SuperH processors), and *x32* (for amd64/x86_64 CPUs using 32-bit pointers) is currently underway.

Il supporto per l'architettura *m68k* è stato abbandonato nel rilascio Etch (Debian 4.0) perché non soddisfaceva i criteri definiti dai Debian Release Manager. Questa architettura copre macchine Amiga e ATARI con processore Motorola 680x0, dove $x \geq 2$, con MMU. Tuttavia il port è ancora attivo e disponibile per l'installazione anche se non fa parte del rilascio stable ufficiale e potrebbe essere riattivato nei rilasci futuri.

Il supporto per *hppa* (macchine PA-RISC della Hewlett-Packard) e *alpha* (sistemi Alpha Compaq/Digital) è stato abbandonato nel rilascio Squeeze (Debian 6.0) per ragioni simili. Anche *arm* è stato abbandonato in questo rilascio, dato che è stato reso obsoleto dall'architettura *armel*.

Support for the 32-bit *s390* port (*s390*) was discontinued and replaced with *s390x* in Jessie (Debian 8). In addition, the ports to IA-64 and Sparc had to be removed from this release due to insufficient developer support.

Per ulteriori informazioni sui port disponibili vedere le pagine dei port sul sito web (<http://www.debian.org/ports/>).

Per ulteriori informazioni su come fare l'avvio, partizionare le proprie unità, abilitare i dispositivi PCMCIA (PC Card) e per questioni simili si seguano le istruzioni date nel manuale di installazione, che è disponibile dal sito WWW di Debian all'indirizzo <http://www.debian.org/releases/stable/installmanual>.

4.2 Quali kernel può eseguire Debian GNU/Linux?

Beside Linux, Debian provides a complete, binary distribution for the following operating system kernels:

- FreeBSD: provided through the *kfreebsd-amd64* and *kfreebsd-i386* ports, for 64-bit PCs and 32-bit PCs respectively. These ports were first released in Debian 6.0 Squeeze as a *technology preview*. However they were not part of the Debian 8 Jessie release.

In aggiunta a questi, sono in fase di lavorazione i seguenti adattamenti:

- *avr32*, port per architettura RISC a 32 bit di Atmel,
- *hurd-i386*, un port per PC a 32 bit. Questo port userà GNU Hurd, il nuovo sistema operativo che il gruppo GNU sta creando,
- *sh*, port per processori Hitachi SuperH.

Ci sono stati tentativi di fare il port della distribuzione per il kernel NetBSD, fornendo *netbsd-i386* (per PC a 32 bit) e *netbsd-alpha* (per macchine Alpha) ma questi port non sono mai stati rilasciati e sono al momento abbandonati.

Per ulteriori informazioni sui port disponibili vedere le pagine dei port sul sito web (<http://www.debian.org/ports/>).

4.3 Quanto è compatibile Debian con le altre distribuzioni Linux?

Gli sviluppatori Debian comunicano con i produttori delle altre distribuzioni Linux nel tentativo di mantenere la compatibilità binaria tra le varie distribuzioni Linux. La maggior parte dei prodotti commerciali per Linux funziona bene sotto Debian come sul sistema su cui è stata compilata.

Debian GNU/Linux aderisce al Linux Filesystem Hierarchy Standard (<http://www.pathname.com/fhs/>) (Standard per la gerarchia del file system Linux). Comunque, c'è spazio per interpretazioni diverse di alcune regole all'interno di questo standard, così ci possono essere piccole differenze tra un sistema Debian e altri sistemi Linux.

Debian GNU/Linux supporta il software sviluppato per la Linux Standard Base (<http://www.linuxbase.org/>). La LSB è una specifica per permettere allo stesso pacchetto binario di essere usato su diverse distribuzioni. Il rilascio Debian Etch è certificato in base alla LSB Release 3.1, vedere la pagina web della Linux Foundation Certification (<https://www.linux-foundation.org/lsb-cert/productdir.php>). La discussione e il coordinamento degli sforzi fatti per assicurare che Debian soddisfi i requisiti della Linux Standard Base avvengono sulla mailing list *debian-lsb* (<http://lists.debian.org/debian-lsb/>).

4.4 Quanto è compatibile Debian con altri sistemi Unix a livello di codice sorgente?

Per la maggior parte delle applicazioni il codice sorgente di Linux è compatibile con altri sistemi Unix. Supporta quasi tutto ciò che è disponibile per i sistemi Unix basati su System V ed i sistemi liberi e commerciali derivati da BSD. Tuttavia nel mondo Unix una tale affermazione non ha quasi nessun valore perché non c'è modo di provarla. Nell'area di sviluppo del software è richiesta una completa compatibilità piuttosto della compatibilità nella "maggioranza" dei casi. Così, anni fa, è sorta la necessità della presenza di standard e al giorno d'oggi POSIX.1 (IEEE Standard 1003.1-1990) è uno dei principali standard per la compatibilità del codice sorgente nei sistemi operativi simil-Unix.

Linux si propone di aderire al POSIX.1, ma gli standard POSIX hanno un costo elevato e la certificazione di POSIX.1 (e FIPS 151-2) è abbastanza onerosa; questo ha reso più difficile agli sviluppatori Linux il lavorare per una completa conformità a POSIX. Il costo della certificazione rende improbabile che Debian ottenga una certificazione di conformità ufficiale anche se passasse completamente la suite di validazione. (La suite di validazione è ora liberamente disponibile, così è previsto che più persone lavoreranno sui problemi relativi a POSIX.1.)

L'Unifix GmbH (Braunschweig, Germania) ha sviluppato un sistema Linux che è stato certificato come conforme a FIPS 151-2 (un sovrainsieme di POSIX.1). Questa tecnologia era disponibile nella distribuzione di Unifix chiamata Unifix Linux 2.0 e nel Linux-FT della Lasermoon.

4.5 Si possono usare i pacchetti Debian (file “.deb”) sul proprio sistema Linux RedHat/Slackware/...? Si possono usare i pacchetti RedHat (file “.rpm”) sul proprio sistema Debian GNU/Linux?

Differenti distribuzioni Linux usano differenti formati dei pacchetti e differenti programmi di gestione dei pacchetti.

Probabilmente si può: È disponibile un programma per spaccettare un pacchetto Debian su un sistema Linux che è stato creato da una distribuzione “straniera”, e generalmente funziona, nel senso che quei file verranno spaccettati. Anche l’inverso è probabilmente vero, cioè un programma che apre un pacchetto RedHat o Slackware su di un sistema basato su Debian GNU/Linux probabilmente riuscirà a spaccettare il pacchetto e a mettere la maggior parte dei file nelle directory di destinazione appropriate. Questo è in gran parte una conseguenza dell’esistenza dello standard Linux Filesystem Hierarchy (e della diffusa conformità ad esso). Il pacchetto Alien (<http://packages.debian.org/alien>) viene usato per fare conversioni tra i diversi formati dei pacchetti.

Probabilmente non si vuole: La maggior parte dei gestori dei pacchetti scrive alcuni file di amministrazione quando viene usata per spaccettare un archivio. Questi file di amministrazione sono generalmente non standardizzati. Quindi, aprire un pacchetto Debian su un sistema “straniero” avrà effetti imprevedibili (certamente non utili) sul gestore dei pacchetti di quel sistema. Similmente, le utilità di un’altra distribuzione possono riuscire ad aprire i loro archivi su sistemi Debian, ma probabilmente causeranno dei guasti al sistema di gestione dei pacchetti Debian al momento di aggiornare o rimuovere alcuni pacchetti, o anche semplicemente nel riportare esattamente quali pacchetti sono presenti sul sistema.

Un metodo migliore: Il Linux File System Standard (e quindi Debian GNU/Linux) richiede che le sottodirectory che si trovano sotto `/usr/local/` siano interamente sottoposte alla discrezione dell’utente. Quindi, gli utenti possono spaccettare pacchetti “stranieri” dentro queste directory e poi possono gestire individualmente la loro configurazione, aggiornamento e rimozione.

4.6 Come installare un programma non-Debian?

I file sotto la directory `/usr/local/` non sono sotto il controllo del sistema di gestione dei pacchetti Debian. Quindi è buona abitudine mettere il codice sorgente del proprio programma in `/usr/local/src/`. Per esempio, si potrebbero estrarre i file da un pacchetto che si chiama “pippo.tar” dentro la directory `/usr/local/src/pippo`. Dopo averli compilati, si mettano i file binari in `/usr/local/bin/`, le librerie in `/usr/local/lib/` e i file di configurazione in `/usr/local/etc/`.

Se i propri programmi o i file devono essere realmente messi in altre directory, si potrebbe sempre memorizzarli in `/usr/local/` e creare i collegamenti simbolici appropriati dalla posizione richiesta alla loro collocazione in `/usr/local/`; per esempio, si potrebbe creare il collegamento

```
ln -s /usr/local/bin/pippo /usr/bin/pippo
```

In ogni caso, se si ottiene un pacchetto il cui copyright permette la redistribuzione, si dovrebbe considerare di trasformarlo in un pacchetto Debian e caricarlo per il sistema Debian. Le linee guida per diventare uno sviluppatore sono incluse nel manuale Debian Policy (si veda ‘Quale altra documentazione esiste su e per un sistema Debian?’ a pagina 53).

4.7 Perché non posso compilare programmi che richiedono libtermcap?

Debian usa il database `terminfo` e la libreria di funzioni per interfaccia a terminale `ncurses`, piuttosto del database `termcap` e della libreria `termcap`.

La libreria a basso livello `terminfo` è stata separata dalla libreria `ncurses` generica creando un nuovo pacchetto chiamato `libtinfo5`. `libncurses` ha il link a `libtinfo`, perciò gli utenti che stanno compilando dei programmi che richiedono una qualche conoscenza delle interfacce dei terminali dovrebbero sostituire i riferimenti a `libtermcap` con i riferimenti a `libncurses`.

Per supportare i file binari per i quali è stato già fatto il link con la libreria `termcap` e per i quali non si hanno i sorgenti, si dovranno trovare i sorgenti della libreria e compilarseli da soli. Debian forniva in passato un pacchetto chiamato `termcap-compat` che conteneva sia `libtermcap.so.2` sia `/etc/termcap`, ma è stato rimosso dalla distribuzione nel 2005.

4.8 Perché non posso installare AccelX?

AccelX usa la libreria `termcap` per l'installazione. Si veda 'Perché non posso compilare programmi che richiedono `libtermcap`?' nella pagina precedente più sopra.

Capitolo 5

Software disponibile nel sistema Debian

5.1 Quali tipi di applicazioni e software di sviluppo sono disponibili per Debian GNU/Linux?

Come la maggior parte delle distribuzioni Linux, Debian GNU/Linux fornisce:

- le principali applicazioni GNU per lo sviluppo del software, la manipolazione dei file e l'elaborazione testi, inclusi gcc, g++, make, texinfo, Emacs, la shell Bash e numerose utilità Unix aggiornate;
- Perl, Python, Tcl/Tk e vari programmi correlati, moduli e librerie per ognuno di essi;
- TeX(LaTeX) e Lyx, dvips, Ghostscript;
- il sistema a finestre Xorg, che fornisce un'interfaccia grafica orientata alla rete per Linux ed innumerevoli applicazioni per X inclusi gli ambienti desktop GNOME, KDE e Xfce;
- a full suite of networking applications, including servers for Internet protocols such as HTTP (WWW), FTP, NNTP (news), SMTP and POP (mail) and DNS (name servers); relational databases like PostgreSQL, MySQL; also provided are web browsers including the various Mozilla products,
- a complete set of office applications, including the LibreOffice productivity suite, Gnumeric and other spreadsheets, WYSIWYG editors, calendars.

More than 42304 packages, ranging from news servers and readers to sound support, FAX programs, database and spreadsheet programs, image processing programs, communications, net, and mail utilities, Web servers, and even ham-radio programs are included in the distribution. Other 696 software suites are available as Debian packages, but are not formally part of Debian due to license restrictions.

5.2 Chi ha scritto tutto questo software?

Per ogni pacchetto gli *autori* del programma (o dei programmi) sono menzionati nel file `/usr/doc/PACCHETTO/copyright`, dove PACCHETTO è il nome del pacchetto.

I *manutentori* che impacchettano questo software per il sistema Debian GNU/Linux sono elencati nel file di controllo Debian (si veda 'Cos'è un file di controllo Debian?' a pagina 28) che è fornito con ogni pacchetto. Anche il changelog Debian, in `/usr/share/doc/PACCHETTO/changelog.Debian.gz`, nomina coloro che hanno lavorato alla creazione del pacchetto Debian.

5.3 Come si può ottenere una lista aggiornata dei programmi che sono stati impacchettati per Debian?

Una lista completa è disponibile in tutti i mirror Debian (<http://www.debian.org/distrib/ftplist>), nel file `indices/Maintainers`. Quel file contiene i nomi dei pacchetti e i nomi e gli indirizzi di posta elettronica dei rispettivi manutentori.

L'interfaccia WWW ai pacchetti Debian (<http://packages.debian.org/>) riassume in modo pratico i pacchetti in ciascuna delle circa trenta "sezioni" dell'archivio Debian.

5.4 Come si può installare un ambiente di sviluppo per compilare pacchetti?

Se si desidera compilare pacchetti nel proprio sistema Debian è necessario avere un ambiente di sviluppo base, incluso un compilatore C/C++ ed alcuni altri pacchetti essenziali. Per installare questo ambiente è sufficiente installare il pacchetto `build-essential`: esso è un metapacchetto o pacchetto segnaposto che dipende dagli strumenti standard di sviluppo che sono necessari per compilare un pacchetto Debian.

È comunque possibile che qualche software possa aver bisogno di software aggiuntivo per essere ricompilato, inclusi header di libreria o strumenti aggiuntivi come `autoconf` o `gettext`. Debian fornisce molti degli strumenti necessari per compilare ulteriore software in forma di pacchetti Debian.

Scoprire precisamente quale software sia necessario può tuttavia essere problematico, a meno che non si desideri ricompilare pacchetti Debian. Quest'ultimo compito è piuttosto facile, dato che i pacchetti ufficiali devono includere un elenco del software aggiuntivo (oltre ai pacchetti in `build-essential`) necessario per compilare il pacchetto; questo elenco è noto come `Build-Dependencies` (dipendenze di compilazione). Per installare tutti i pacchetti necessari per compilare un dato pacchetto sorgente e quindi effettivamente compilarlo basta eseguire:

```
# apt-get build-dep pippo
# apt-get source --build pippo
```

Si noti che se si desiderano compilare i kernel Linux distribuiti da Debian si dovrà installare anche il pacchetto `kernel-package`. Per maggiori informazioni si veda 'Quali strumenti fornisce Debian per costruire kernel personalizzati?' a pagina 45.

5.5 Cosa manca da Debian GNU/Linux?

There is a list of packages which still need to be packaged for Debian, the Work-Needing and Prospective Packages list (<http://www.debian.org/devel/wnpp/>).

For more details about adding missing things, see 'How can I become a Debian member/Debian developer?' a pagina 57.

5.6 Perché ricevo il messaggio "ld: cannot find -lpippo" quando compilo i programmi? Perché non c'è alcun file libpippo.so nei pacchetti delle librerie Debian?

La Debian Policy richiede che tali collegamenti simbolici (a `libpippo.so.x.y.z` o simile) siano posti in un pacchetto di sviluppo separato. Questi pacchetti di solito si chiamano `libpippo-dev` o `libpippoX-dev` (ammettendo che il pacchetto libreria si chiami `libpippoX` e che X sia un numero intero).

5.7 Debian supporta Java? (E come?)

Diverse implementazioni *libere* della tecnologia Java sono disponibili come pacchetti Debian e forniscono sia Java Development Kit sia ambienti Runtime. Usando Debian si possono scrivere programmi Java, eseguirli e farne il debug.

Running a Java applet requires a web browser with the capability to recognize and execute it. Several web browsers available in Debian, such as Mozilla or Konqueror, support Java plug-ins that enable running Java applets within them.

Si faccia riferimento alla Debian Java FAQ (<http://www.debian.org/doc/manuals/debian-java-faq/>) per maggiori informazioni.

5.8 How can I check that I am using a Debian system, and what version it is?

Per assicurarsi che il proprio sistema sia stato installato dai veri dischi di base di Debian usare il comando

```
lsb_release -a
```

Esso visualizzerà il nome della distribuzione (nel campo Distributor ID) e la versione del sistema (nei campi Release e Codename). Questo è un esempio di esecuzione in un sistema Debian:

```
$ lsb_release -a
No LSB modules are available.
Distributor ID: Debian
Description:   Debian GNU/Linux 7.4 (wheezy)
Release:      7.4
Codename:     wheezy
```

Si può anche controllare l'esistenza del file `/etc/debian_version` che contiene una singola voce di una riga che fornisce il numero di versione del rilascio, come definito dal pacchetto `base-files`.

Gli utenti dovrebbero comunque essere consci del fatto che il sistema Debian consiste di molte parti, ognuna delle quali può essere aggiornata (quasi) indipendentemente. Ogni "rilascio" Debian è formato da un contenuto ben definito e invariato. Gli aggiornamenti sono disponibili separatamente. Per una descrizione su un'unica riga dello stato di installazione del pacchetto `pippo`, si usi il comando `dpkg --get-architecture pippo`. Per una descrizione più prolissa, usare:

```
dpkg --get-architecture pippo
```

Per vedere le versioni di tutti i pacchetti installati eseguire:

```
dpkg -l
```

Note that the existence of the program `dpkg` shows that you should be able to install Debian packages on your system. However, since the program has been ported to many other operating systems and architectures, this is no longer a reliable method of determining if a system is Debian GNU/Linux.

5.9 Come supporta Debian le lingue diverse dall'inglese?

- Debian GNU/Linux è distribuita con mappe di tastiera per quasi due dozzine di tastiere e con utilità (nel pacchetto `kbd`) per installare, vedere e modificare le tabelle.

L'installazione chiede all'utente di specificare la tastiera che usa.

- Nearly all of the software in Debian supports UTF-8 as character set. Legacy character sets, such as ISO-8859-1 or ISO-8859-2, should be considered obsolete.
- Currently, support for German-, Spanish-, French-, Hungarian-, Italian-, Japanese-, Korean-, Dutch-, Polish-, Portuguese-, Russian-, Turkish-, and Chinese-language manual pages is provided through the `manpages-LANG` packages (where LANG is the two-letter ISO country code). To access an NLS manual page, the user must set the shell `LC_MESSAGES` variable to the appropriate string.

Per esempio, nel caso delle pagine di manuale in lingua italiana, `LC_MESSAGES` deve essere impostato a "italian". Il programma `man` cercherà poi le pagine di manuale in italiano sotto la directory `/usr/share/man/it/`.

5.10 Dove è ezmlm/djbdns/qmail?

Una volta Dan J. Bernstein distribuiva tutto il software che ha scritto (<http://cr.yip.to/software.html>) con una licenza restrittiva che non permetteva di distribuire i binari modificati. Tuttavia, nel novembre 2007, Bernstein ha detto "[...] ho deciso di rendere tutto il mio software a venire e [...] tutto quello passato di pubblico dominio". Si veda FAQ dai distributori (<http://cr.yip.to/distributors.html>) per i suoi termini di distribuzione.

Al momento dell'ultima stesura di questo documento (marzo 2016) `ezmlm-idx` è disponibile solamente in `experimental` (il pacchetto `mlmmj` è simile ed è fornito con Debian `jessie`); il pacchetto `djbdns` è disponibile solamente in `sid` (`unstable`), si possono vedere Bug nr.516394 (<https://bugs.debian.org/516394>) e Bug nr.796118 (<https://bugs.debian.org/796118>) per i dettagli e vedere `dbndns` per un'alternativa simile; il software `publicfile` continua ad essere software non libero, è disponibile un pacchetto `publicfile-installer` nella sezione `contrib` di Debian.

Altro software di Dan J. Bernstein (`qmail`, `daemontools`, `ucspi-tcp`) viene fornito in Debian.

5.11 Dove è un riproduttore per Flash (SWF)?

Debian fornisce sia `gnash` sia `swfdec`: due riproduttori liberi per filmati SWF.

5.12 Dove è Google Earth?

Google Earth is available for GNU/Linux from Google's web site, but not only it is not Free Software, but is completely undistributable by a third party. However, `googleearth-package` (in the contrib-section) might be helpful in using this software.

5.13 Dove è il software VoIP?

Per il Voice Over IP vengono usati principalmente due protocolli aperti: SIP e H.323. Entrambi sono implementati da una grande varietà di software presente nella sezione main di Debian. `ekiga` è uno dei client più popolari.

5.14 Ho una scheda di rete wireless che non funziona con Linux. Cosa fare?

Comprarne una che funzioni :)

In alternativa, usare `ndiswrapper` per usare un driver per Windows (se lo si ha) sul proprio sistema Linux. Si veda la pagina su `ndiswrapper` nel Debian Wiki (<http://wiki.debian.org/NdisWrapper>) per maggiori informazioni.

Capitolo 6

Gli archivi FTP Debian

6.1 Quante distribuzioni di Debian ci sono?

Esistono tre distribuzioni principali: la distribuzione “stable” (stabile), la distribuzione “testing” (in test) e la distribuzione “unstable” (instabile). La distribuzione “testing” è a volte congelata, “frozen” (si veda ‘Cosa dire di “testing”? Come viene “congelata”?’ a pagina 23). Oltre a queste c’è la distribuzione “oldstable” (che è semplicemente quella che ha preceduto “stable”) e quella “experimental”.

Experimental is used for packages which are still being developed, and with a high risk of breaking your system. It’s used by developers who’d like to study and test bleeding edge software. Users shouldn’t be using packages from there, because they can be dangerous and harmful even for the most experienced people.

Per un aiuto al momento di scegliere una distribuzione Debian, si veda ‘Scegliere una distribuzione Debian’ a pagina 7.

6.2 Cosa sono tutti quei nomi come etch, lenny, ecc.?

Sono solo “nomi in codice”. Quando una distribuzione Debian è in fase di sviluppo non ha un numero di versione ma un nome in codice. Lo scopo di questi nomi in codice è di rendere più semplice la creazione di mirror delle distribuzioni Debian (se una directory reale come `unstable` cambiasse improvvisamente il nome in `stable`, moltissima roba dovrebbe essere inutilmente scaricata di nuovo).

Attualmente, `stable` è un collegamento simbolico a `jessie` (ovvero Debian GNU/Linux 8) e `testing` è un collegamento simbolico a `stretch`. Questo significa che `jessie` è la distribuzione stable attuale e che `stretch` è la distribuzione testing attuale.

`unstable` è un collegamento simbolico permanente a `sid`, dato che `sid` è sempre la distribuzione unstable (si veda ‘Cosa dire a proposito di “sid”?’ nella pagina seguente).

6.2.1 Quali altri nomi in codice sono stati usati in passato?

Aside `jessie` and `stretch`, other codenames that have been already used are: `buzz` for release 1.1, `rex` for release 1.2, `bo` for releases 1.3.x, `hamm` for release 2.0, `slink` for release 2.1, `potato` for release 2.2, `woody` for release 3.0, `sarge` for release 3.1, `etch` for release 4.0, `lenny` for release 5.0, `squeeze` for release 6.0, and `wheezy` for release 7.

6.2.2 Da dove derivano questi nomi in codice?

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 pecora,
- `hamm` era il salvadanaio a forma di maialino,

- *slink* (Slinky Dog) era il cane giocattolo,
- *potato* era ovviamente Mr. Potato,
- *woody* era il cowboy,
- *sarge* era il sergente dell'Armata Verde,
- *etch* era la lavagna giocattolo (Etch-a-Sketch),
- *lenny* era il binocolo giocattolo,
- *squeeze* era il nome degli alieni con tre occhi,
- *wheezy* era il pinguino di gomma con un farfallino rosso,
- *jessie* era la cowgirl che faceva lo yodel,
- *stretch* era la piovra giocattolo di gomma con ventose sulle sue otto lunghe braccia,
- *sid* era il bambino cattivo della casa accanto che distruggeva tutti i giocattoli.

La decisione (<https://lists.debian.org/debian-devel/1996/06/msg00515.html>) di usare i nomi in Toy Story è stata presa (<https://lists.debian.org/debian-user/1997/04/msg00011.html>) da Bruce Perens che, all'epoca, era il Debian Project Leader e stava contemporaneamente lavorando alla Pixar, la compagnia che ha prodotto i film.

6.3 Cosa dire a proposito di “sid”?

sid o *unstable* è il posto in cui la maggior parte dei pacchetti viene inizialmente caricata. Non sarà mai direttamente rilasciata, perché i pacchetti che devono essere rilasciati devono prima essere inclusi in *testing*, per poter essere rilasciati in *stable* più tardi. *sid* contiene pacchetti sia per architetture rilasciate che non.

Anche il nome “*sid*” proviene dal film d'animazione “Toy Story”: Sid era il bambino della porta accanto che distruggeva i giocattoli :-)

1

6.4 Cosa contiene la directory *stable*?

- *stable/main/*: questa directory contiene i pacchetti che costituiscono formalmente il rilascio più recente del sistema Debian GNU/Linux.

Tutti questi pacchetti sono conformi alle DFSG - Linee guida Debian per il software libero (http://www.debian.org/social_contract#guidelines) e sono tutti liberamente utilizzabili e distribuibili.

- *stable/non-free/*: questa directory contiene i pacchetti la cui distribuzione è limitata in modo tale da richiedere che i distributori prendano attentamente in considerazione i loro requisiti specifici relativi al copyright.

Per esempio, alcuni pacchetti hanno licenze che ne vietano la distribuzione commerciale. Altri possono essere redistribuiti, ma sono di fatto shareware e non software libero. Le licenze di ognuno di questi pacchetti devono essere studiate e possibilmente negoziate prima che tali pacchetti possano essere inclusi in qualsiasi redistribuzione (per esempio, in un CD-ROM).

- *stable/contrib/*: questa directory contiene i pacchetti che sono di per sé liberi in base alle DFSG e *liberamente distribuibili*, ma dipendono in qualche modo da un pacchetto che *non* è liberamente distribuibile ed è quindi disponibile solo nella sezione *non-free*.

¹Quando l'attuale *sid* non esisteva, l'organizzazione del sito FTP aveva un grande difetto: c'era l'assunto che quando un'architettura veniva creata nell'attuale *unstable*, sarebbe stata rilasciata quando quella distribuzione diventava la nuova *stable*. Per molte architetture questo non è vero, con il risultato che quelle directory dovevano essere spostate al momento del rilascio. Ciò era poco pratico, poiché lo spostamento avrebbe divorato grosse quantità di banda. Gli amministratori dell'archivio hanno aggirato questo problema per diversi anni collocando i binari delle architetture non rilasciate in una directory speciale chiamata “*sid*”. Per quelle architetture non ancora rilasciate, al primo rilascio c'era un collegamento da *stable* a *sid* e da quel momento in poi venivano create all'interno dell'albero *unstable* come di norma. Tutto ciò era motivo di confusione per gli utenti. Con l'avvento dei pool di pacchetti (si veda ‘Cosa c'è nella directory *pool*?’ a pagina 24), i pacchetti binari cominciarono ad essere immagazzinati in una posizione canonica nel pool, indipendentemente dalla distribuzione, così il rilascio di una distribuzione non determina più grande dispendio di banda sui mirror (c'è, ovviamente, un notevole graduale consumo di banda durante la fase di sviluppo).

6.5 Cosa contiene la distribuzione testing?

I pacchetti vengono inseriti nella directory “testing” dopo aver subito un periodo di test in unstable.

They must be in sync on all architectures where they have been built and mustn't have dependencies that make them uninstalleable; they also need to have fewer release-critical bugs than the versions currently in unstable. This way, we hope that 'testing' is always close to being a release candidate.

Maggiori informazioni sullo stato di “testing” in generale e dei singoli pacchetti sono disponibili su <http://www.debian.org/devel/testing>.

6.5.1 Cosa dire di “testing”? Come viene “congelata”?

When the “testing” distribution is mature enough, the release manager starts ‘freezing’ it. The normal propagation delays are increased to ensure that as few new bugs as possible from “unstable” enter “testing”.

Dopo un po', la distribuzione “testing” diventa realmente “congelata” (frozen). Ciò significa che tutti i nuovi pacchetti da mettere in “testing” sono trattenuti, a meno che non contengano le soluzioni a bug critici per il rilascio. La distribuzione “testing” può anche rimanere in questo stato di “surgelamento” durante i cosiddetti “cicli di test”, quando il rilascio è imminente.

Quando un rilascio diventa “congelato”, “unstable” tende a congelarsi parzialmente anch'essa. Ciò avviene perché gli sviluppatori sono restii a caricare software radicalmente nuovo in unstable, nel caso che il software congelato in testing necessiti di aggiornamenti minori e per risolvere bug critici per il rilascio che impediscono a testing di diventare “stable”.

Viene tenuto un registro dei bug nella distribuzione “testing” che possono impedire ad un pacchetto di essere rilasciato, o dei bug che possono impedire l'intero rilascio. Per i dettagli si vedano le informazioni sull'attuale rilascio testing (<http://www.debian.org/releases/testing/>).

Una volta che il numero dei bug si abbassa sotto i valori massimi accettabili, la distribuzione “congelata” viene dichiarata “stable” e rilasciata con un proprio numero di versione.

Il conteggio più importante per i bug è quello dei bug critici per il rilascio (“Release Critical”), che può essere seguito nella Pagina dello stato dei bug critici per il rilascio (<http://bugs.debian.org/release-critical/>). Un comune obiettivo di rilascio è NoRCBugs (<http://wiki.debian.org/ReleaseGoals/NoRCBugs>) che significa che la distribuzione non dovrebbe avere alcun bug di gravità critica, grave o seria. L'elenco completo dei problemi considerati critici può essere trovato nel documento della politica RC (http://release.debian.org/testing/rc_policy.txt).

Ad ogni nuovo rilascio, la precedente distribuzione “stable” diventa obsoleta e viene spostata in archivio. Per maggiori informazioni si veda l'archivio Debian (<http://www.debian.org/distrib/archive>).

6.6 Cosa contiene la distribuzione unstable?

La directory “unstable” contiene un'istantanea del sistema attualmente in via di sviluppo. Gli utenti sono i benvenuti ad usare e testare questi pacchetti, ma sono avvisati riguardo il loro stato di preparazione. Il vantaggio di usare la distribuzione “unstable” è che si è sempre aggiornati con la più recente produzione di software in GNU/Linux, ma se si rompe, i cocci sono vostri :-)

Anche in “unstable” ci sono le sottodirectory main, contrib e non-free, separate con lo stesso criterio adottato in “stable”.

6.7 Cosa sono tutte quelle directory negli archivi FTP Debian?

Il software che è stato impacchettato per Debian GNU/Linux è disponibile in uno dei diversi alberi di directory in ogni sito mirror Debian.

La directory `dists` è l'abbreviazione di “distribuzioni” ed è il percorso canonico per accedere ai rilasci (e pre-rilasci) Debian attualmente disponibili.

La directory `pool` contiene i pacchetti effettivi, si veda ‘Cosa c'è nella directory `pool`?’ nella pagina seguente.

Ci sono queste directory aggiuntive:

/tools/: utilità DOS per creare dischi di avvio, partizionare il proprio disco, comprimere/decomprimere file e avviare Linux.

/doc/: la documentazione di base di Debian, come queste FAQ, le istruzioni del sistema di segnalazione dei bug, ecc.

/indices/: vari indici del sito (il file `Maintainers` e i file `override`).

/project/: per la maggior parte materiale solo per gli sviluppatori e file vari.

6.8 Cosa sono tutte quelle directory dentro a `dists/stable/main`?

All'interno di ciascuno degli alberi di directory principali², ci sono tre insiemi di sottodirectory contenenti file indice.

C'è un gruppo di sottodirectory `binary-qualcosa` che contengono i file indice per i pacchetti binari di ciascuna architettura disponibile, per esempio `binary-i386` per i pacchetti che si possono eseguire su macchine PC Intel x86 o `binary-sparc` per i pacchetti da eseguire su SPARCStation Sun.

L'elenco completo delle architetture disponibili per ciascun rilascio è disponibile alla pagina web dei rilasci (<http://www.debian.org/releases/>). Per il rilascio attuale, si veda "Su quali architetture o sistemi hardware funziona Debian?" a pagina 13.

I file indice in `binary-*` si chiamano `Packages(.gz, .bz2)` e contengono un riassunto di ciascun pacchetto binario che è incluso in quella distribuzione. I pacchetti binari effettivi risiedono nella directory `pool` di livello più alto.

Inoltre esiste una sottodirectory chiamata `source/` che contiene i file indice dei pacchetti sorgenti inclusi nella distribuzione. Il file indice si chiama `Sources(.gz, .bz2)`.

Da ultimo, ma non per importanza, c'è un gruppo di sottodirectory pensate per i file indice del sistema di installazione: sono in `debian-installer/binary-architettura`.

6.9 Dove è il codice sorgente?

Viene fornito il codice sorgente per tutto ciò che è nel sistema Debian. Inoltre, i termini di licenza della maggior parte dei programmi *richiedono* che il codice venga distribuito insieme ai programmi o che un'offerta di fornire il codice sorgente li accompagni.

Il codice sorgente viene distribuito nella directory `pool` (si veda "Cosa c'è nella directory `pool`?" in questa pagina) insieme con tutte le directory dei binari specifiche per le architetture. Per ottenere il codice sorgente senza la necessità di avere familiarità con la struttura dell'archivio FTP Debian, si provi un comando come `apt-get source nome-del-miopacchetto`.

A causa di restrizioni nelle licenze, il codice sorgente può essere o meno disponibile per i pacchetti nelle aree «contrib» e «non-free», che non fanno formalmente parte del sistema Debian. In alcuni casi possono essere distribuiti solo dei «frammenti binari» («binary blob») senza sorgenti (vedere ad esempio `firmware-misc-nonfree`); in altri casi la licenza proibisce la distribuzione di binari precompilati, ma permette quella di pacchetti di codice sorgente che l'utente può compilare localmente (vedere `broadcom-sta-dkms`).

6.10 Cosa c'è nella directory `pool`?

I pacchetti vengono tenuti in un grosso "pool", strutturato in base ai nomi dei pacchetti sorgente. Per rendere il tutto usabile, il pool è suddiviso in sezioni ("main", "contrib" e "non-free") e in base alla prima lettera del nome dei pacchetti sorgente. Queste directory contengono diversi file: i pacchetti binari per ciascuna architettura e i pacchetti sorgente da cui sono stati generati i pacchetti binari.

Si può scoprire dove ciascun pacchetto è situato eseguendo un comando come `apt-cache showsrc nome-del-miopacchetto` e guardando la riga "Directory:". Per esempio, i pacchetti `apache` sono immagazzinati in `pool/main/a/apache/`.

Inoltre, poiché ci sono così tanti pacchetti `lib*`, questi vengono trattati in maniera particolare: per esempio, i pacchetti `libpaper` sono immagazzinati in `pool/main/libp/libpaper/`.

3

²`dists/stable/main`, `dists/stable/contrib`, `dists/stable/non-free` e `dists/unstable/main/`, ecc.

³Storicamente, i pacchetti erano tenuti nella sottodirectory di `dists` corrispondente alla distribuzione di cui facevano parte. Questo si è rivelato fonte di vari problemi, come un grosso consumo di banda dei mirror ogni volta che venivano fatti dei cambiamenti di grossa portata; la cosa è stata risolta con l'introduzione dei pool di pacchetti. Le directory `dists` vengono ancora utilizzate per i file indice usati da programmi come `apt`.

6.11 Cos'è "incoming"?

Dopo che uno sviluppatore carica un pacchetto, questo resta per un po' nella directory "incoming" prima che ne venga controllata la genuinità e che venga accettato nell'archivio.

Normalmente nessuno dovrebbe installare cose da questo posto. Comunque, per alcuni rari casi di emergenza, la directory incoming è disponibile su <http://incoming.debian.org/>. Si possono scaricare i pacchetti manualmente, controllare la firma GPG e i codici di controllo MD5 nei file .changes e .dsc, e poi installarli.

6.12 Come impostare un proprio repository usabile con apt?

Se si sono compilati alcuni pacchetti Debian privati che si desiderano installare usando gli strumenti standard per la gestione dei pacchetti Debian, si può impostare un proprio archivio di pacchetti usabile con apt. Questo è utile anche se si desiderano condividere i propri pacchetti Debian quando questi non sono distribuiti dal progetto Debian. Le istruzioni per farlo sono contenute nel Wiki Debian (<https://wiki.debian.org/HowToSetupADebianRepository>).

Capitolo 7

Fondamenti sul sistema di gestione dei pacchetti Debian

Questo capitolo tratta alcuni aspetti dettagliati a basso livello della gestione dei pacchetti di Debian. Se si è principalmente interessati nell'uso degli strumenti relativi, si salti ai capitoli 'Gli strumenti di gestione dei pacchetti Debian' a pagina 35 e 'Mantenere il sistema Debian aggiornato' a pagina 41.

7.1 Cosa è un pacchetto Debian?

I pacchetti generalmente contengono tutti quei file necessari a implementare una serie di comandi o funzionalità correlate. Ci sono due tipi di pacchetti Debian:

- *Binary packages*, which contain executables, configuration files, man/info pages, copyright information, and other documentation. These packages are distributed in a Debian-specific archive format (see 'Qual è il formato di un pacchetto binario Debian?' nella pagina successiva); they are usually characterized by having a '.deb' file extension. Binary packages can be unpacked using the Debian utility `dpkg` (possibly via a frontend like `aptitude`); details are given in its manual page.
- *Source packages*, which consist of a `.dsc` file describing the source package (including the names of the following files), a `.orig.tar.gz` file that contains the original unmodified source in gzip-compressed tar format and usually a `.diff.gz` file that contains the Debian-specific changes to the original source. The utility `dpkg-source` packs and unpacks Debian source archives; details are provided in its manual page. (The program `apt-get` can be used as a frontend for `dpkg-source`.)

L'installazione di software attraverso il sistema dei pacchetti usa "dipendenze" che vengono progettate con cura dai manutentori dei pacchetti. Queste dipendenze sono documentate nel file `control` associato ad ogni pacchetto. Per esempio, il pacchetto contenente il compilatore C GNU (`gcc`) "dipende" dal pacchetto `binutils` che include il linker e l'assemblatore. Se un utente tenta di installare `gcc` senza avere prima installato `binutils`, il sistema di gestione dei pacchetti (`dpkg`) invia un messaggio di errore avvertendo che necessita anche di `binutils`, e ferma l'installazione di `gcc`. (Questa funzione può essere tuttavia aggirata dall'utente tenace, si veda `dpkg(8)`.) Per maggiori informazioni, si veda 'Cosa significa dire che un pacchetto *Dipende da*, *Raccomanda*, *Suggerisce*, *Va in conflitto con*, *Sostituisce*, *Rompe* o *Fornisce* un altro pacchetto?' a pagina 31 più sotto.

Gli strumenti Debian per i pacchetti possono essere usati per:

- manipolare e gestire pacchetti o parti di pacchetti,
- amministrare le deviazioni locali dei file in un pacchetto,
- aiutare gli sviluppatori nella costruzione degli archivi dei pacchetti e
- aiutare gli utenti nell'installazione di pacchetti che sono in un sito FTP remoto.

7.2 Qual è il formato di un pacchetto binario Debian?

Un “pacchetto” Debian, o un file archivio Debian, contiene i file eseguibili, le librerie e la documentazione associati ad una particolare suite di programmi o ad un particolare gruppo di programmi correlati. Normalmente, un file archivio Debian ha un nome che termina in `.deb`.

I dettagli di questo formato per i pacchetti binari Debian sono descritti nella pagina di manuale `deb(5)`. Questo formato interno è soggetto a modifiche (tra diversi rilasci principali di Debian GNU/Linux), quindi usare sempre `dpkg-deb(1)` se si devono effettuare manipolazioni a basso livello di file `.deb`.

7.3 Perché i nomi dei pacchetti Debian sono così lunghi?

Il nome dei pacchetti binari Debian segue la seguente convenzione: `<pippo>_<NumeroVersione>-<NumeroRevisioneDebian>_<ArchitetturaDebian>.deb`

Note that `f00` is supposed to be the package name. Checking the package name associated with a particular Debian archive file (`.deb` file) can be done in one of these ways:

- Esaminando il file “Packages” nella directory dove era stato archiviato su un archivio FTP Debian. Questo file contiene una sezione che descrive ogni pacchetto; il primo campo in ogni sezione è il nome formale del pacchetto.
- Utilizzando il comando `dpkg --info pippo_VVV-RRR_AAA.deb` (dove VVV, RRR e AAA sono rispettivamente la versione, la revisione e l’architettura del pacchetto in questione). Questo mostra, tra le altre cose, il nome del pacchetto corrispondente al file archivio spaccettato.

La parte VVV è il numero di versione specificato dallo sviluppatore originale. Qui non ci sono standard, per cui il numero di versione può avere formati differenti come “19990513” e “1.3.8pre1”.

La parte RRR è il numero di revisione Debian e viene specificata dallo sviluppatore Debian (o il singolo utente se sceglie di crearsi il pacchetto da sé). Questo numero corrisponde al livello di revisione del pacchetto Debian, quindi un nuovo livello di revisione significa solitamente modifiche nel Makefile Debian (`debian/rules`), nel file di controllo Debian (`debian/control`), negli script di installazione o rimozione (`debian/p*`) oppure nei file di configurazione utilizzati con il pacchetto.

The AAA component identifies the processor for which the package was built. This is commonly `amd64`, which refers to AMD64, Intel 64 or VIA Nano chips. For other possibilities review Debian’s FTP directory structure at ‘Cosa sono tutte quelle directory negli archivi FTP Debian?’ a pagina 23. For details, see the description of “Debian architecture” in the manual page `dpkg-architecture(1)`.

7.4 Cos’è un file di controllo Debian?

Le specifiche riguardanti il contenuto di un file di controllo Debian sono fornite nel manuale Debian Policy, sezione 5, si veda ‘Quale altra documentazione esiste su e per un sistema Debian?’ a pagina 53.

Brevemente, un esempio di file di controllo è mostrato di seguito per il pacchetto Debian `hello`:

```
Package: hello
Version: 2.9-2+deb8u1
Architecture: amd64
Maintainer: Santiago Vila <sanvila@debian.org>
Installed-Size: 145
Depends: libc6 (>= 2.14)
Conflicts: hello-traditional
Breaks: hello-debhelper (<< 2.9)
Replaces: hello-debhelper (<< 2.9), hello-traditional
Section: devel
Priority: optional
Homepage: http://www.gnu.org/software/hello/
Description: example package based on GNU hello
 The GNU hello program produces a familiar, friendly greeting. It
 allows non-programmers to use a classic computer science tool which
 would otherwise be unavailable to them.
.
 Seriously, though: this is an example of how to do a Debian package.
 It is the Debian version of the GNU Project’s ‘hello world’ program
 (which is itself an example for the GNU Project).
```

Il campo `Package` fornisce il nome del pacchetto. Questo è il nome attraverso il quale il pacchetto può essere manipolato con gli strumenti per i pacchetti ed è solitamente simile, ma non necessariamente identico, alla prima parte della stringa del nome del file archivio Debian.

Il campo `Version` fornisce sia il numero della versione dello sviluppatore originale, sia (nell'ultima parte) il livello di revisione del pacchetto Debian di questo programma, come spiegato in 'Perché i nomi dei pacchetti Debian sono così lunghi?' nella pagina precedente.

Il campo `Architecture` specifica il chip per il quale questo particolare binario è stato compilato.

Il campo `Depends` fornisce una lista di pacchetti che devono essere installati per poter installare questo pacchetto con successo.

`Installed-Size` indica quanto spazio su disco occuperà il pacchetto installato. È pensato per essere usato dai frontend di installazione in modo da poter mostrare se c'è abbastanza spazio disponibile su disco per installare il programma.

La riga `Section` fornisce la "sezione" in cui questo pacchetto Debian è archiviato presso i siti FTP Debian.

`Priority` indica quanto è importante questo pacchetto per l'installazione, così che i software semi-intelligenti come `apt` o `aptitude` possano ordinare il pacchetto dentro una categoria di, ad esempio, pacchetti installati in modo opzionale. Si veda 'Cosa sono i pacchetti *Essential, Required, Important, Standard, Optional* o *Extra*?' nella pagina seguente.

Il campo `Maintainer` fornisce l'indirizzo di posta elettronica della persona che è attualmente responsabile della manutenzione di questo pacchetto.

Il campo `Description` fornisce un breve riassunto delle funzionalità del pacchetto.

Per ulteriori informazioni riguardo tutti i campi possibili che può avere un pacchetto, si veda il manuale Debian Policy, sezione 5, "Control files and their fields" (vedere 'Quale altra documentazione esiste su e per un sistema Debian?' a pagina 53).

7.5 Cos'è un conffile Debian?

Conffiles è una lista di file di configurazione (solitamente situati in `/etc`) che il sistema di gestione dei pacchetti non sovrascriverà quando il pacchetto viene aggiornato. Questo assicura che i valori locali per il contenuto di questi file vengano preservati, ed è una caratteristica critica che permette l'aggiornamento in loco di pacchetti su un sistema in esecuzione.

Per determinare esattamente quali file sono preservati durante un aggiornamento, si esegua:

```
dpkg --status pacchetto
```

e si guardi alla voce "Conffiles:".

7.6 Cosa sono gli script `preinst`, `postinst`, `prerm` e `postrm` di Debian?

Questi file sono script eseguibili che vengono automaticamente eseguiti prima o dopo l'installazione o la rimozione di un pacchetto. Insieme ad un file chiamato `control`, tutti questi file sono parte della sezione «control» di un file archivio Debian.

I singoli file sono:

preinst This script is executed before the package it belongs to is unpacked from its Debian archive (".deb") file. Many 'preinst' scripts stop services for packages which are being upgraded until their installation or upgrade is completed (following the successful execution of the 'postinst' script).

postinst This script typically completes any required configuration of the package `foo` once `foo` has been unpacked from its Debian archive (".deb") file. Often, 'postinst' scripts ask users for input, and/or warn them that if they accept default values, they should remember to go back and re-configure that package as needed. Many 'postinst' scripts then execute any commands necessary to start or restart a service once a new package has been installed or upgraded.

prerm Questo script tipicamente ferma tutti i demoni associati ad un pacchetto. Viene eseguito prima della rimozione di file associati al pacchetto.

postrm Questo script tipicamente modifica i collegamenti o altri file associati a `pippo`, o rimuove i file creati da quel pacchetto. (Si veda anche 'Cos'è un pacchetto virtuale?' nella pagina seguente.)

Currently all of the control files can be found in the directory `/var/lib/dpkg/info`. The files relevant to package `foo` begin with the name “foo” and have file extensions of “preinst”, “postinst”, etc., as appropriate. The file `foo.list` in that directory lists all of the files that were installed with the package `foo`. (Note that the location of these files is a dpkg internal; you should not rely on it.)

7.7 Cosa sono i pacchetti *Essential*, *Required*, *Important*, *Standard*, *Optional* o *Extra*?

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

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

Comprendono tutti gli strumenti che sono necessari per riparare i difetti di sistema. Questi pacchetti non devono essere rimossi o il proprio sistema potrebbe diventare completamente non funzionante e probabilmente non si riuscirebbe nemmeno ad usare dpkg per rimettere le cose a posto. I sistemi con solo i pacchetti Required sono probabilmente inutilizzabili, ma hanno abbastanza funzionalità per permettere all'amministratore di sistema di avviare ed installare altri programmi.

- **Important (Importante):** pacchetti che si dovrebbero trovare su tutti i sistemi *nix.

Other packages which the system will not run well or be usable without will be here. This does *NOT* include Emacs or X or TeX or any other large application. These packages only constitute the bare infrastructure.

- **Standard** packages are standard on any Linux system, including a reasonably small but not too limited character-mode system. Tools are included to be able to send e-mail (with mutt) and download files from FTP servers.

This is what will be installed by default if users do not select anything else. It does not include many large applications, but it does include the Python interpreter and some server software like OpenSSH (for remote administration) and Exim (for mail delivery, although it can be configured for local delivery only). It also includes some common generic documentation that most users will find helpful.

- **Optional** packages include all those that you might reasonably want to install if you do not know what they are, or that do not have specialized requirements.

Comprendono X, 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 si sa a cosa servono, oppure hanno requisiti speciali che li rendono non adatti come “Optional”.

Se si fa un'installazione Debian predefinita verranno installati sul sistema tutti i pacchetti di priorità **Standard** o maggiore. Se si selezionano attività predefinite si otterranno anche pacchetti con priorità più bassa.

In aggiunta, alcuni pacchetti sono marcati come **Essential (Essenziali)** dato che sono assolutamente necessari per il corretto funzionamento del sistema. Gli strumenti di gestione dei pacchetti si rifiuteranno di rimuoverli.

7.8 Cos'è un pacchetto virtuale?

A virtual package is a generic name that applies to any one of a group of packages, all of which provide similar basic functionality. For example, both the `konqueror` and `firefox-esr` programs are web browsers, and should therefore satisfy any dependency of a program that requires a web browser on a system, in order to work or to be useful. They are therefore both said to provide the “virtual package” called `www-browser`.

Similarly, `exim4` and `sendmail` both provide the functionality of a mail transport agent. They are therefore said to provide the virtual package “mail-transport-agent”. If either one is installed, then any program depending on the installation of a `mail-transport-agent` will be satisfied by the presence of this virtual package.

Debian fornisce un meccanismo in modo che, se più di un pacchetto che fornisce lo stesso pacchetto virtuale è installato su di un sistema, l'amministratore di sistema può allora impostarne uno come pacchetto preferito. Il relativo comando è `update-alternatives` ed è descritto più in dettaglio in ‘Ad alcuni utenti piace mawk, ad altri gawk; ad alcuni piace vim, ad altri elvis; ad alcuni piace trn, ad altri tin; come supporta Debian le diversità?’ a pagina 50.

7.9 Cosa significa dire che un pacchetto *Dipende da*, *Raccomanda*, *Suggerisce*, *Va in conflitto con*, *Sostituisce*, *Rompe* o *Fornisce* un altro pacchetto?

Il sistema dei pacchetti Debian ha una serie di “dipendenze” dei pacchetti che sono pensate per indicare (con un singolo termine) il livello a cui, su un dato sistema, un determinato Programma A può funzionare indipendentemente dall’esistenza di un Programma B.

- Il pacchetto A *dipende* dal Pacchetto B se B deve essere assolutamente installato per eseguire A. In alcuni casi, A dipende non solo da B, ma da una versione di B. In questo caso la dipendenza dalla versione è solitamente un limite inferiore, nel senso che A dipende da qualsiasi versione di B più recente di quella specificata.
- Il pacchetto A *raccomanda* il Pacchetto B se il manutentore del pacchetto giudica che la maggior parte degli utenti non vorrebbe A senza le funzionalità fornite da B.
- Il pacchetto A *suggerisce* il Pacchetto B se B contiene file attinenti alle funzioni di A (e che solitamente le migliorano).
- Il pacchetto A *va in conflitto* con il Pacchetto B quando A non è in grado di funzionare se B è installato sul sistema. Molto spesso i conflitti si hanno quando A contiene dei file che rappresentano dei miglioramenti di quelli in B. Spesso “va in conflitto” è associato a “sostituisce”.
- Il pacchetto A *sostituisce* il Pacchetto B quando i file installati da B vengono rimossi e (in alcuni casi) sovrascritti dai file di A.
- Il pacchetto A *rompe* il pacchetto B quando non è possibile configurare simultaneamente entrambi i pacchetti su un sistema. Il sistema di gestione dei pacchetti si rifiuta di installare uno se l’altro è già installato e configurato sul sistema.
- Il pacchetto A *fornisce* il Pacchetto B quando tutti i file e le funzionalità di B sono incorporate in A. Questo meccanismo permette agli utenti con limitato spazio su disco rigido di avere solo la parte del pacchetto A realmente necessaria per loro.

More detailed information on the use of each of these terms can be found in the Debian Policy manual, section 7.2, “Binary Dependencies”, see ‘Quale altra documentazione esiste su e per un sistema Debian?’ a pagina 53.

7.10 Cosa significa Pre-Depends?

“Pre-Depends” is a special dependency. In the case of most packages, `dpkg` will unpack the archive file of a package (i.e., its `.deb` file) independently of whether or not the files on which it depends exist on the system. Simplistically, unpacking means that `dpkg` will extract the files from the archive file that were meant to be installed on your file system, and put them in place. If those packages *depend* on the existence of some other packages on your system, `dpkg` will refuse to complete the installation (by executing its “configure” action) until the other packages are installed.

Tuttavia, per alcuni pacchetti, `dpkg` si rifiuta persino di spaccettarli finché certe dipendenze non vengono risolte. Tali pacchetti si dice che “Pre-dipendono” dalla presenza di altri pacchetti. Il progetto Debian aveva fornito questo meccanismo per supportare un aggiornamento sicuro di sistemi dal formato `a.out` al formato ELF, dove l’ordine in cui i pacchetti venivano spaccettati era critico. Esistono altre situazioni di aggiornamenti estesi in cui questo metodo è utile, per esempio pacchetti con priorità richiesta e la loro dipendenza da `libc`.

Come sopra, informazioni più dettagliate al riguardo possono essere reperite nel manuale Debian Policy.

7.11 Cosa significano *sconosciuto*, *installa*, *rimuovi*, *elimina* e *blocca* nello stato di un pacchetto?

Queste etichette dei «desiderata» indicano il volere dell’utente riguardo ad un pacchetto (come indicato direttamente dall’utente nell’esecuzione di `dpkg/apt/aptitude`).

I loro significati sono:

- *sconosciuto*: l’utente non ha mai indicato se vuole il pacchetto;
- *installa*: l’utente vuole che il pacchetto sia installato o aggiornato;

- `remove` - the user wants the package removed, but does not want to remove any existing configuration file.
- `elimina`: l'utente vuole che il pacchetto sia completamente rimosso, compresi i file di configurazione;
- `blocca`: l'utente non vuole che il pacchetto sia processato, ovvero vuole mantenere la versione attuale con lo stato attuale, qualunque essi siano.

7.12 Come si blocca un pacchetto?

Esistono tre modi per bloccare (hold) pacchetti, con `dpkg`, `apt` o `aptitude`.

Con `dpkg`, si deve solo esportare la lista delle selezioni dei pacchetti con:

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

Poi modificare il file risultante `selezione.txt`, cambiando la riga contenente il pacchetto che si desidera bloccare, per esempio `libc6`, da questo:

```
libc6                                install
```

a questo:

```
libc6                                hold
```

Salvare il file e ricaricarlo nel database di `dpkg` con:

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

Con `apt` si può bloccare un pacchetto usando

```
apt-mark hold nome_pacchetto
```

e rimuovere il blocco con

```
apt-mark unhold nome_pacchetto
```

Con `aptitude`, si può bloccare un pacchetto usando

```
aptitude hold nome_pacchetto
```

e rimuovere il blocco con

```
aptitude unhold nome_pacchetto
```

7.13 Come si installa un pacchetto sorgente?

I pacchetti sorgente Debian non possono realmente venire "installati", vengono solo spaccettati in qualsiasi directory in cui si vogliono compilare i pacchetti binari che producono.

Source packages are distributed on most of the same mirrors where you can obtain the binary packages. If you set up your APT's `sources.list` (5) to include the appropriate "deb-src" lines, you'll be able to easily download any source package by running

```
apt-get source pippo
```

To help you in actually building the source package, Debian source packages provide the so-called build-dependencies mechanism. This means that the source package maintainer keeps a list of other packages that are required to build their package. To see how this is useful, run

```
apt-get build-dep pippo
```

prima di compilare il sorgente.

7.14 Come si compilano pacchetti binari da un pacchetto sorgente?

Il metodo consigliato è usare vari strumenti wrapper. Verrà mostrato come farlo usando gli strumenti `devscripts`. Installare questo pacchetto se non è ancora stato fatto.

Ora, come prima cosa recuperare il pacchetto sorgente:

```
apt-get source pippo
```

e spostarsi nell'albero dei sorgenti:

```
cd pippo-*
```

Poi installare le dipendenze di compilazione necessarie (se ce ne sono):

```
sudo apt-get build-dep pippo
```

Creare quindi una versione dedicata della propria compilazione (in modo da non confondersi successivamente quando Debian stessa rilascerà una nuova versione):

```
dch -l local 'Bla bla bla'
```

Da ultimo compilare il proprio pacchetto:

```
debuild -us -uc
```

Se tutto ha funzionato a dovere, dovrebbe essere ora possibile installare il proprio pacchetto eseguendo

```
sudo dpkg -i ../*.deb
```

Se si preferisce fare le cose a mano, e non si vuole usare `devscripts`, seguire la procedura seguente:

Si avrà bisogno di tutti i file `pippo_*.dsc`, `pippo_*.tar.gz` e `pippo_*.diff.gz` per compilare i sorgenti (nota: non c'è nessun `.diff.gz` per alcuni pacchetti che sono nativi di Debian).

Una volta che li si ha ('Come si installa un pacchetto sorgente?' nella pagina precedente), e si ha il pacchetto `dpkg-dev` installato, il seguente comando

```
dpkg-source -x pippo_versione-revisione.dsc
```

estrarrà il pacchetto in una directory chiamata `pippo-versione`.

If you just want to compile the package, you may `cd` into the `foo-version` directory and issue the command

```
dpkg-buildpackage -rfakeroot -b
```

per compilare il pacchetto (si noti che questo richiede anche il pacchetto `fakeroot`), e poi

```
dpkg -i ../pippo_versione-revisione_arch.deb
```

per installare il pacchetto appena compilato.

7.15 Come ci si creano pacchetti Debian da sé?

For a more detailed description on this, read the New Maintainers' Guide, available in the `maint-guide` package or at <http://www.debian.org/doc/devel-manuals#maint-guide>, or the Guide for Debian Maintainers, available in the `debmake-doc` package or at <https://www.debian.org/doc/devel-manuals#debmake-doc>.

Capitolo 8

Gli strumenti di gestione dei pacchetti Debian

8.1 Quali programmi fornisce Debian per gestire i suoi pacchetti?

Esistono diversi strumenti che vengono usati per gestire i pacchetti Debian, dalle interfacce grafiche o testuali agli strumenti a basso livello usati per installare i pacchetti. Tutti gli strumenti disponibili, per funzionare correttamente, si appoggiano sugli strumenti di più basso livello e vengono elencati qui in ordine di complessità decrescente.

È importante capire che gli strumenti di gestione dei pacchetti di più alto livello, come `aptitude` o `synaptic` si appoggiano su `apt` che si appoggia, a sua volta, su `dpkg` per gestire i pacchetti nel sistema.

See Chapter 2. Debian package management (<http://www.debian.org/doc/manuals/debian-reference/ch02.en.html>) of the Debian reference (<http://www.debian.org/doc/manuals/debian-reference/>) for more information about the Debian package management utilities. This document is available in various languages and formats, see the Debian Reference entry in the DDP Users' Manuals overview (<http://www.debian.org/doc/user-manuals#quick-reference>).

8.1.1 dpkg

Questo è il principale programma di gestione dei pacchetti. `dpkg` può essere invocato con molte opzioni. Quelli che seguono sono alcuni degli usi comuni.

- Scoprire tutte le opzioni: `dpkg --help`.
- Stampare il file di controllo (ed altre informazioni) di un specifico pacchetto: `dpkg --info pippo_VVV-RRR.deb`.
- Installare un pacchetto (incluso lo spaccettamento e la configurazione) nel file system dell'hard disk: `dpkg --install pippo_VVV-RRR.deb`.
- Spaccettare (ma non configurare) un archivio Debian sul file system dell'hard disk: `dpkg --unpack pippo_VVV-RRR.deb`. Si noti che questa operazione *non* lascia necessariamente il pacchetto in uno stato utilizzabile; alcuni file possono necessitare di ulteriori adattamenti per funzionare correttamente. Questo comando rimuove qualsiasi versione già installata del programma ed esegue lo script `preinst` (si veda 'Cosa sono gli script `preinst`, `postinst`, `prepm` e `postrm` di Debian?' a pagina 29) associato al pacchetto.
- Configurare un pacchetto che è già stato spaccettato: `dpkg --configure pippo`. Tra le altre cose, questa azione esegue lo script `postinst` (si veda 'Cosa sono gli script `preinst`, `postinst`, `prepm` e `postrm` di Debian?' a pagina 29) associato al pacchetto. Aggiorna inoltre i file elencati in `conffiles` per questo pacchetto. Si faccia attenzione che l'operazione "configure" prende come argomento il nome del pacchetto (ad esempio, `pippo`), *non* il nome del file archivio Debian (ad esempio, `pippo_VVV-RRR.deb`).
- Estrarre un unico file chiamato "ciccia" (o un gruppo di file chiamati "ciccia*") da un archivio Debian: `dpkg --fsys-tarfile pippo_VVV-RRR.deb | tar -xf - 'ciccia*'`.
- Rimuovere un pacchetto (ma non i suoi file di configurazione): `dpkg --remove pippo`.
- Rimuovere un pacchetto (inclusi i suoi file di configurazione): `dpkg --purge pippo`.
- Elencare lo stato d'installazione dei pacchetti contenenti la stringa (o l'espressione regolare) "pippo*": `dpkg --list 'pippo*'`.

8.1.2 APT

APT is the *Advanced Package Tool*, an advanced interface to the Debian packaging system which provides the `apt-get` program. It provides commandline tools for searching and managing packages, and for querying information about them, as well as low-level access to all features of the `libapt-pkg` library. For more information, see the User's Guide in `/usr/share/doc/apt-doc/guide.html/index.html` (you will have to install the `apt-doc` package).

Starting with Debian Jessie, some frequently used `apt-get` and `apt-cache` commands have an equivalent via the new `apt` binary. This means some popular commands like `apt-get update`, `apt-get install`, `apt-get remove`, `apt-cache search`, or `apt-cache show` now can also be called simply via `apt`, say `apt update`, `apt install`, `apt remove`, `apt search`, or `apt show`. The following is an overview of the old and new commands:

```
apt-get update          -> apt update
apt-get upgrade        -> apt upgrade
apt-get dist-upgrade   -> apt full-upgrade
apt-get install package -> apt install pacchetto
apt-get remove package -> apt remove pacchetto
apt-get autoremove     -> apt autoremove
apt-cache search string -> apt search stringa
apt-cache policy package -> apt list -a pacchetto
apt-cache show package -> apt show pacchetto
apt-cache showpkg package -> apt show -a pacchetto
```

Lo strumento `apt` unisce funzionalità di `apt-get` e `apt-cache` e, in modo predefinito, ha un formato di output colorato più elegante che lo rende più piacevole per l'utente. Per l'uso in script o casi d'uso avanzati `apt-get` rimane preferibile o necessario.

`apt-get` fornisce un metodo semplice per recuperare e installare i pacchetti da fonti multiple usando la riga di comando. A differenza di `dpkg`, `apt-get` non capisce i file `.deb`, funziona con i nomi appropriati dei pacchetti e può solo installare gli archivi `.deb` da una fonte specificata in `/etc/apt/sources.list`. `apt-get` chiama `dpkg` direttamente dopo aver scaricato gli archivi `.deb`¹ dalle fonti configurate.

Quelli che seguono sono alcuni dei modi comuni di usare `apt-get`.

- To update the list of packages known by your system, you can run:

```
apt update
```

(lo si dovrebbe eseguire regolarmente per aggiornare il proprio elenco di pacchetti)

- Per installare il pacchetto *pippo* e tutte le sue dipendenze, si esegua:

```
apt install pippo
```

- Per rimuovere il pacchetto *pippo* dal sistema, si esegua:

```
apt remove pippo
```

- Per rimuovere il pacchetto *pippo* e i suoi file di configurazione dal sistema, si esegua:

```
apt purge pippo
```

- To list all packages for which newer versions are available, run:

```
apt list --upgradable
```

- Per aggiornare tutti i pacchetti sul sistema (senza installare pacchetti aggiuntivi o rimuovere pacchetti), si esegua:

```
apt upgrade
```

- Per aggiornare tutti i pacchetti sul sistema e, se necessario per l'aggiornamento di un pacchetto, installare pacchetti aggiuntivi o rimuovere pacchetti, si esegua:

```
apt full-upgrade
```

¹Si noti che ci sono port che rendono disponibile questo strumento con altri sistemi di gestione dei pacchetti, come il gestore dei pacchetti di Red Hat, noto anche come `rpm`.

(Il comando `upgrade` mantiene un pacchetto nella sua versione installata obsoleta se l'aggiornamento necessita dell'installazione di un pacchetto aggiuntivo per soddisfare una nuova dipendenza. Il comando `full-upgrade` è meno conservativo.)

Note that you must be logged in as root to perform any commands that modify packages.

Note that `apt-get` now also installs recommended packages as default, and thanks to its robustness it's the preferred program for package management from console to perform system installation and major system upgrades.

La suite di strumenti `apt` include anche lo strumento `apt-cache` per interrogare l'elenco dei pacchetti. Lo si può usare per trovare i pacchetti che forniscono una funzionalità specifica, attraverso l'uso di interrogazioni con testo semplice o espressioni regolari e con interrogazioni sulle dipendenze nel sistema di gestione dei pacchetti. Alcuni dei modi comuni di usare `apt-cache` sono i seguenti.

- Per trovare pacchetti la cui descrizione contiene *parola*:

```
apt search parola
```

- Per stampare le informazioni dettagliate su un pacchetto:

```
apt show pacchetto
```

- Per stampare i pacchetti da cui dipende un dato pacchetto:

```
apt-cache depends pacchetto
```

- To print detailed information on the versions available for a package and the packages that reverse-depend on it:

```
apt-cache showpkg pacchetto
```

Per maggiori informazioni, installare il pacchetto `apt` e leggere `apt(8)`, `apt-get(8)`, `sources.list(5)` e installare il pacchetto `apt-doc` e leggere `/usr/share/doc/apt-doc/guide.html/index.html`.

8.1.3 aptitude

`aptitude` è un gestore di pacchetti per sistemi Debian GNU/Linux che fornisce un frontend per l'infrastruttura di gestione dei pacchetti `apt`. `aptitude` è un'interfaccia testuale che usa la libreria `curses`. Le azioni possono essere effettuate da un'interfaccia visuale o dalla riga di comando.

`aptitude` può essere usato per effettuare compiti di gestione dei pacchetti in modo veloce e semplice. Permette all'utente di visualizzare l'elenco dei pacchetti e di eseguire compiti di gestione dei pacchetti come installazioni, aggiornamenti e rimozioni.

`aptitude` fornisce le funzionalità di `apt-get`, oltre a molte funzionalità aggiuntive:

- `aptitude` offre un facile accesso a tutte le versioni di un pacchetto;
- `aptitude` rende facile tenere traccia del software obsoleto elencandolo in "Pacchetti obsoleti e creati localmente";
- `aptitude` include un sistema piuttosto potente per cercare particolari pacchetti e limitare i pacchetti visualizzati. Gli utenti che hanno familiarità con `mutt` impareranno presto ad usarlo, dato che la sintassi per le espressioni regolari è stata ispirata da quella di `mutt`.
- `aptitude` può essere usato per installare le attività predefinite disponibili. Per maggiori informazioni si veda 'tasksel' nella pagina seguente.
- `aptitude` in full screen mode has `su` functionality embedded and can be run by a normal user. It will call `su` (and ask for the root password, if any) when you really need administrative privileges.

Si può usare `aptitude` attraverso un'interfaccia visiva (si esegua semplicemente `aptitude`) o direttamente dalla riga di comando. La sintassi usata per la riga di comando è molto simile a quella usata con `apt-get`. Per esempio, per installare il pacchetto `pippo`, si può eseguire `aptitude install pippo`.

Note that `aptitude` is the preferred program for daily package management from the console.

For more information, read the manual page `aptitude(8)` and install the `aptitude-doc` package.

8.1.4 synaptic

`synaptic` is a graphical package manager. It enables you to install, upgrade and remove software packages in a user friendly way. Along with most of the features offered by `aptitude`, it also has a feature for editing the list of used repositories, and supports browsing all available documentation related to a package. See the Synaptic Website (<http://www.nongnu.org/synaptic/>) for more information.

8.1.5 tasksel

Quando si desidera fare una specifica attività può essere difficile trovare l'insieme appropriato di pacchetti che soddisfano le proprie necessità. Gli sviluppatori Debian hanno definito delle attività (task): un'attività è una raccolta di diversi pacchetti Debian individuali tutti relativi ad una specifica attività. Le attività possono essere installate tramite il programma `tasksel` o con `aptitude`.

Typically, the Debian installer will automatically install the task associated with a standard system and a desktop environment. The specific desktop environment installed will depend on the CD/DVD media used, most commonly it will be the GNOME desktop (`gnome-desktop` task). Also, depending on your selections throughout the installation process, tasks might be automatically installed in your system. For example, if you selected a language other than English, the task associated with it will be installed automatically too and if the installer recognises you are installing on a laptop system the `laptop` task will also be installed.

8.1.6 Altri strumenti per la gestione dei pacchetti

`dpkg-deb`

This program manipulates Debian archive (`.deb`) files. Some common uses are:

- Scoprire tutte le opzioni: `dpkg-deb --help`.
- Determinare quali file sono contenuti in un file archivio Debian: `dpkg-deb --contents pippo_VVV-RRR.deb`
- Estrarre i file contenuti in un archivio Debian dentro una directory specificata dall'utente: `dpkg-deb --extract pippo_VVV-RRR.deb tmp` estrae ognuno dei file in `pippo_VVV-RRR.deb` nella directory `tmp/`. Questo è comodo per esaminare il contenuto di un pacchetto in una directory localizzata, senza installare il pacchetto nel file system `root`.
- Estrarre i file delle informazioni di controllo da un pacchetto: `dpkg-deb --control pippo_VVV-RRR.deb tmp`

Si noti che ogni pacchetto che è stato semplicemente spaccettato usando `dpkg-deb --extract` non sarà installato correttamente, si dovrebbe invece usare `dpkg --install`.

Maggiori informazioni sono fornite nella pagina di manuale `dpkg-deb(1)`.

8.2 Debian afferma di essere in grado di aggiornare un programma in esecuzione; come viene fatto?

Il kernel (file system) nei sistemi Debian GNU/Linux supporta la sostituzione dei file anche quando sono in uso.

È fornito anche un programma chiamato `start-stop-daemon` che viene usato per avviare i demoni all'avvio o fermarli quando il runlevel cambia (per esempio, da multiutente a singolo utente o all'arresto). Lo stesso programma viene usato dagli script di installazione quando viene installato un nuovo pacchetto contenente un demone, per fermare i demoni in esecuzione e riavviarli se necessario.

8.3 Come si può sapere quali pacchetti sono già installati su un sistema Debian?

Per conoscere lo stato di tutti i pacchetti installati su di un sistema Debian, si esegua il comando

```
dpkg --list
```

Questo stampa un riassunto di una riga per ogni pacchetto, dando un simbolo di stato di due lettere (spiegato nell'intestazione), il nome del pacchetto, la versione *installata* e una breve descrizione.

To learn the status of packages whose names match any pattern beginning with “foo”, run the command:

```
dpkg --list 'pippo*'
```

Per ottenere un rapporto più prolisso per un particolare pacchetto si esegua il comando:

```
dpkg --status nomepacchetto
```

8.4 How do I display the files of an installed package?

Per elencare tutti i file forniti dal pacchetto `pippo` installato si esegua il comando

```
dpkg --listfiles pippo
```

Notare che i file creati dagli script di installazione non vengono mostrati.

8.5 Come posso scoprire quale pacchetto ha prodotto un particolare file?

Per identificare il pacchetto che ha prodotto il file chiamato `pippo` si esegua uno dei comandi:

- `dpkg --search pippo`

Questo cerca `pippo` nei pacchetti installati. (È (attualmente) equivalente a cercare nella directory `/var/lib/dpkg/info/` tutti i file aventi l'estensione `.list` e a sistemare l'output per stampare i nomi di tutti i pacchetti e le deviazioni che lo contengono.)

Un'alternativa più veloce è lo strumento `dlocate`.

```
dlocate -S pippo
```

- `zgrep pippo Contents-ARCH.gz`

Questo cerca i file contenenti la sottostringa `pippo` nel loro nome di percorso completo. I file `Contents-ARCH.gz` (dove `ARCH` rappresenta l'architettura desiderata) risiedono nelle directory principali dei pacchetti (`main`, `non-free`, `contrib`) sui siti FTP Debian (cioè in `/debian/dists/jessie`). Un file `Contents` si riferisce solo ai pacchetti nell'albero delle sottodirectory dove esso risiede. Quindi, un utente potrebbe dover cercare in più file `Contents` per trovare il pacchetto contenente il file `pippo`.

Questo metodo ha il vantaggio rispetto a `dpkg --search` di trovare i file nei pacchetti che non sono attualmente installati sul proprio sistema.

- `apt-file search pippo`

If you install the `apt-file` package, similar to the above, it searches files which contain the substring or regular expression `foo` in their full path names. The advantage over the example above is that there is no need to retrieve the `Contents-ARCH.gz` files as it will do this automatically for all the sources defined in `/etc/apt/sources.list` when you run (as root) `apt-file update`.

8.6 Why is 'foo-data' not removed when I uninstall 'foo'? How do I make sure old unused library-packages get purged?

Some packages are split in program (“foo”) and data (“foo-data”) (or in “foo” and “foo-doc”). This is true for many games, multimedia applications and dictionaries in Debian and has been introduced since some users might want to access the raw data without installing the program or because the program can be run without the data itself, making “foo-data” optional.

Situazioni simili si verificano quando si ha a che fare con le librerie: generalmente vengono installate dato che i pacchetti che contengono applicazioni dipendono da esse. Quando il pacchetto dell'applicazione viene completamente eliminato, il

pacchetto di libreria può rimanere sul sistema. Oppure, quando il pacchetto dell'applicazione non dipende più da libdb4.2, ad esempio, ma da libdb4.3, il pacchetto libdb4.2 può rimanere quando il pacchetto dell'applicazione viene aggiornato.

In these cases, 'foo-data' doesn't depend on 'foo', so when you remove the 'foo' package it will not get automatically removed by most package management tools. The same holds true for the library packages. This is necessary to avoid circular dependencies. However, if you use `apt-get` (see 'APT' a pagina 36) or `aptitude` (see 'aptitude' a pagina 37) as your package management tool, they will track automatically installed packages and give the possibility to remove them, when no packages making use of them remain in your system.

Capitolo 9

Mantenere il sistema Debian aggiornato

One of Debian's goals is to provide a consistent upgrade path and a secure upgrade process. We always do our best to make upgrading to new releases a smooth procedure. In case there's some important note to add to the upgrade process, the packages will alert the user, and often provide a solution to a possible problem.

You should also read the Release Notes document that describes the details of specific upgrades. It is available on the Debian website at <http://www.debian.org/releases/stable/releasenotes> and is also shipped on the Debian CDs, DVDs and Blu-ray discs.

9.1 Come si può mantenere il proprio sistema Debian aggiornato?

Si può eseguire semplicemente un ftp anonimo ad un archivio Debian, poi esplorare le directory fino a quando si trova il file desiderato, scaricarlo e finalmente installarlo usando `dpkg`. Si noti che `dpkg` installa i file aggiornati al volo, anche su un sistema in esecuzione. Qualche volta, un pacchetto revisionato richiede l'installazione di una rinnovata versione di un altro pacchetto, in questo caso l'installazione fallisce fino a quando/a meno che l'altro pacchetto non venga installato.

Molte persone trovano questo approccio troppo dispendioso in termini di tempo, dato che Debian si evolve molto rapidamente: tipicamente, sono caricati ogni settimana una dozzina o più di nuovi pacchetti. Questo numero aumenta appena prima di un nuovo rilascio principale. Per gestire questa valanga, molte persone preferiscono usare un metodo più automatico. Sono disponibili diversi pacchetti a questo scopo:

9.1.1 aptitude

`aptitude` è il gestore di pacchetti raccomandato per i sistemi Debian GNU/Linux ed è descritto in 'aptitude' a pagina 37.

Prima di poter usare `aptitude` per effettuare un aggiornamento, è necessario modificare il file `/etc/apt/sources.list` per configurarlo. Se si desidera aggiornare all'ultima versione stabile di Debian, probabilmente si vorrà usare una fonte simile a quella seguente:

```
http://ftp.us.debian.org/debian stable main contrib
```

Si può sostituire `ftp.us.debian.org` (il mirror negli Stati Uniti) con il nome di un mirror Debian più veloce e più vicino. Per maggiori informazioni si veda l'elenco dei mirror su <http://www.debian.org/mirror/list>.

Or you can use the redirector service `httpredir.debian.org` which aims to solve the problem of choosing a Debian mirror. It uses the geographic location of the user and other information to choose the best mirror that can serve the files. To take advantage of it use a source like this one:

```
http://httpredir.debian.org/debian stable main contrib
```

Ulteriori dettagli a questo proposito possono essere trovati nella pagina di manuale `sources.list(5)`.

To update your system from the command line, run

```
aptitude update
```

seguito da

```
aptitude full-upgrade
```

Rispondere a tutte le domande eventualmente poste e il sistema sarà aggiornato.

Notare che `aptitude` non è lo strumento raccomandato per aggiornare da un rilascio di Debian GNU/Linux ad un altro. Usare invece `apt-get`. Per gli aggiornamenti tra i rilasci si dovrebbero leggere le Note di rilascio (<http://www.debian.org/releases/stable/releasenotes>). Tale documento descrive in dettaglio i passi raccomandati per aggiornare da rilasci precedenti, così come problemi conosciuti che devono essere presi in considerazione prima dell'aggiornamento.

Per i dettagli, vedere la pagina di manuale `aptitude(8)` e il file `/usr/share/aptitude/README`.

9.1.2 apt-get e apt-cdrom

Un'alternativa a `aptitude` è `apt-get` che è uno strumento a riga di comando basato su APT (descritto in precedenza in 'APT' a pagina 36).

`apt-get`, lo strumento a riga di comando basato su APT per gestire i pacchetti, fornisce un modo semplice e sicuro per installare ed aggiornare pacchetti.

Per usare `apt-get`, modificare il file `/etc/apt/sources.list` per configurarlo, proprio come per 'aptitude' nella pagina precedente.

Poi eseguire

```
apt-get update
```

seguito da

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

Rispondere a tutte le domande eventualmente poste e il sistema sarà aggiornato. Vedere anche la pagina di manuale `apt-get(8)`, oltre a 'APT' a pagina 36.

Se si vogliono usare CD/DVD/BD per installare i pacchetti, si può usare `apt-cdrom`. Per dettagli, vedere nelle Note di rilascio (Release Notes) la sezione sull'aggiunta di fonti APT da supporti ottici.

Notare che quando si ottengono e si installano i pacchetti, li si ha ancora nella gerarchia di directory in `/var`. Per evitare che la propria partizione strabordi, ricordarsi di eliminare i file in più usando `apt-get clean` e `apt-get autoclean`, oppure di spostarli in qualche altro posto (consiglio: si usi `apt-move`).

9.1.3 mirror

Questo script in Perl, e il suo (opzionale) programma di gestione chiamato `mirror-master`, può essere usato per prelevare parti di alberi di directory specificate dall'utente da uno specifico host *via* FTP anonimo.

`mirror` è particolarmente utile per scaricare grandi quantità di software. Dopo che i file sono stati scaricati da un sito per la prima volta, viene memorizzato nell'host locale un file chiamato `.mirrorinfo`. `mirror` tiene automaticamente traccia dei cambiamenti al file system remoto, paragonando questo file ad un file simile sul sistema remoto e scarica solo i file cambiati.

The `mirror` program is generally useful for updating local copies of remote directory trees. The files fetched need not be Debian files. (Since `mirror` is a Perl script, it can also run on non-Unix systems.) Though the `mirror` program provides mechanisms for excluding files with names matching user-specified strings, this program is most useful when the objective is to download whole directory trees, rather than selected packages.

9.2 È necessario entrare in modalità singolo-utente per aggiornare un pacchetto?

No. I pacchetti possono essere aggiornati al volo, anche su sistemi in esecuzione. Debian ha un programma `start-stop-daemon` che viene invocato per fermare e poi riavviare i processi in esecuzione, se necessario, durante l'aggiornamento di un pacchetto.

9.3 È necessario tenere tutti questi file di archivio .deb sul disco?

No. Se i file sono stati scaricati nel proprio disco, allora possono essere rimossi dal sistema dopo l'installazione dei pacchetti; lo si può fare ad esempio eseguendo `aptitude clean`.

9.4 How can I keep a log of the packages I added to the system? I'd like to know when upgrades and removals have occurred and on which packages!

Passare l'opzione `--log` a `dpkg` fa sì che `dpkg` registri i cambiamenti di stato, gli aggiornamenti e le azioni. Registra sia l'invocazione di `dpkg` (ad esempio

```
2005-12-30 18:10:33 install hello 1.3.18 2.1.1-4
```

) sia il risultato (ad esempio

```
2005-12-30 18:10:35 status installed hello 2.1.1-4
```

) If you'd like to log all your `dpkg` invocations (even those done using frontends like `aptitude`), you could add

```
log /var/log/dpkg.log
```

to your `/etc/dpkg/dpkg.cfg`. Be sure the created logfile gets rotated periodically. If you're using `logrotate`, this can be achieved by creating a file `/etc/logrotate.d/dpkg` with the following lines

```
/var/log/dpkg {  
    missingok  
    notifempty  
}
```

Si possono trovare ulteriori dettagli sul registro di `dpkg` nella pagina di manuale `dpkg(1)`.

`aptitude` tiene un registro delle installazioni, rimozioni e degli aggiornamenti di pacchetti che intende effettuare in `/var/log/aptitude`. Notare che in questo file non sono registrati i *risultati* di queste azioni.

Un altro modo di registrare le proprie azioni è di eseguire la propria sessione all'interno del programma `script(1)`.

9.5 È possibile aggiornare automaticamente il sistema?

Sì. Si può usare `cron-apt`; questo strumento aggiorna il sistema ad intervalli regolari usando un compito `cron`. In modo predefinito aggiorna solamente l'elenco dei pacchetti e scarica i nuovi pacchetti, ma senza installarli.

Nota bene: l'aggiornamento automatico di pacchetti **NON** è raccomandato nei sistemi *testing* o *unstable* dato che potrebbe portare a comportamenti inattesi e rimuovere pacchetti senza preavviso.

9.6 Se si hanno svariate macchine come è possibile scaricare gli aggiornamenti una volta sola?

Se si ha più di una macchina Debian nella propria rete, è utile usare `apt-cacher` per mantenere aggiornati tutti i propri sistemi Debian.

`apt-cacher` riduce i requisiti in termini di banda per i mirror Debian, riducendo la frequenza degli aggiornamenti dei file Packages, Releases e Sources dal backend e scaricando ogni file una singola volta, indipendentemente dalle effettive richieste dal proxy. `apt-cacher` genera automaticamente un mirror HTTP Debian in base alle richieste che passano per il proxy.

Naturalmente si possono ottenere gli stessi benefici se si sta già usando un proxy standard con cache e se tutti i propri sistemi sono configurati per farne uso.

Capitolo 10

Debian e il kernel

10.1 Si può installare e compilare un kernel senza alcun adattamento specifico per Debian?

Sì.

There's only one common catch: the Debian C libraries are built with the most recent *stable* releases of the **kernel** headers. If you happen to need to compile a program with kernel headers newer than the ones from the stable branch, then you should either upgrade the package containing the headers (`linux-libc-dev`), or use the new headers from an unpacked tree of the newer kernel. That is, if the kernel sources are in `/usr/src/linux`, then you should add `-I/usr/src/linux/include/` to your command line when compiling.

10.2 Quali strumenti fornisce Debian per costruire kernel personalizzati?

Users who wish to (or must) build a custom kernel are encouraged to use the Debian package target included with recent versions of the kernel build system. After configuring the kernel, simply run the following command:

```
make deb-pkg
```

The new kernel package will be created in the directory one level above the kernel source tree, and it may be installed using `dpkg -i`.

Gli utenti devono scaricare separatamente il codice sorgente per il kernel più recente (o un kernel a propria scelta) dall'archivio del proprio sito Linux preferito, a meno che non sia disponibile un pacchetto `linux-source-versione` (dove *versione* rappresenta la versione del kernel).

10.3 Quali strumenti speciali fornisce Debian per lavorare con i moduli?

A configuration file containing modules to be manually loaded at boot time is kept at `/etc/modules`. However, editing this file is rarely needed.

Other module configuration is kept in the `/etc/modprobe.d/` directory. More information about the format of those files can be found in the `modprobe.conf(5)` manual page.

10.4 Si può disinstallare in sicurezza un vecchio pacchetto kernel e, se sì, come?

Sì. Lo script `linux-image-NNN.preperm` verifica se il kernel che si sta attualmente usando è lo stesso che si sta tentando di disinstallare. Quindi si possono rimuovere i pacchetti delle immagini del kernel non desiderate usando questo comando:

```
dpkg --purge linux-image-NNN
```

(ovviamente sostituire *NNN* con il proprio numero di versione e revisione).

10.5 Where can I get more information about Linux packages for Debian?

Further information is maintained in the Debian Linux Kernel Handbook (<http://kernel-handbook.alioth.debian.org/>).

Capitolo 11

Personalizzare il proprio sistema Debian GNU/Linux

11.1 Come ci si può assicurare che tutti i programmi usino lo stesso formato per la carta?

Si installi il pacchetto `libpaper1` che chiederà il formato carta predefinito per tutto il sistema. Questa impostazione sarà memorizzata nel file `/etc/papersize`.

Gli utenti possono sovrascrivere l'impostazione del formato carta usando la variabile d'ambiente `PAPERSIZE`. Per i dettagli, vedere la pagina di manuale `papersize(5)`.

11.2 Come si può fornire accesso alle periferiche hardware senza compromettere la sicurezza?

Molti file di device nella directory `/dev` appartengono a gruppi predefiniti. Per esempio, `/dev/sr0` appartiene al gruppo `cdrom`.

Se si vuole che un certo utente abbia accesso ad uno di questi device, aggiungere l'utente al gruppo a cui appartiene il device, usare cioè:

```
adduser utente gruppo
```

In questo modo non si dovranno cambiare i permessi dei file di device.

Se lo si fa dall'interno di una shell utente o di un ambiente con interfaccia grafica si deve fare il `logout` e nuovamente il `login` per poter diventare un membro effettivo del gruppo specificato. Per controllare a quali gruppi si appartiene eseguire `groups`.

Si noti che, a partire dall'introduzione di `udev`, se si cambiano i permessi di una periferica hardware questi potrebbero essere modificati all'avvio per alcuni device; se ciò accade alle periferiche hardware a cui si è interessati, sarà necessario modificare in modo appropriato le regole in `/etc/udev`.

11.3 Come caricare un tipo di carattere per la console all'avvio nella maniera Debian?

Il pacchetto `kbd` lo permette: modificare il file `/etc/kbd/config`.

11.4 Come si possono configurare le impostazioni predefinite per l'applicazione di un programma X11?

I programmi X di Debian installano i loro dati di risorsa dell'applicazione nella directory `/etc/X11/app-defaults/`. Se si vogliono personalizzare globalmente le applicazioni X, si mettano le proprie personalizzazioni in questi file. Sono marcati come file di configurazione, quindi il loro contenuto sarà preservato durante gli aggiornamenti.

11.5 Ogni distribuzione sembra avere un metodo di avvio differente. Cosa dire di quello di Debian?

Come tutti gli Unix, Debian si avvia eseguendo il programma `init`¹. Il file di configurazione per `init` (che è `/etc/inittab`) specifica che il primo script da eseguire dovrebbe essere `/etc/init.d/rcS`. Questo script esegue tutti gli script in `/etc/rcS.d/` facendo il fork di sottoprocessi per effettuare un'inizializzazione come verificare e montare i file system, caricare moduli, avviare i servizi di rete, impostare l'orologio ed effettuare altre inizializzazioni.

Dopo aver completato il processo di avvio, `init` esegue tutti gli script di avvio dentro una directory specificata dal livello di esecuzione (runlevel) predefinito (questo runlevel è dato dalla voce `id` in `/etc/inittab`). Come la maggior parte degli Unix* compatibili con il System V, Linux ha 7 runlevel:

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

I sistemi Debian vengono forniti con `id=2`, che indica che il runlevel predefinito sarà "2" quando si entra nello stato multiutente, e che verranno eseguiti gli script in `/etc/rc2.d/`.

Debian usa l'ordine di avvio basato su dipendenze attraverso `insserv`, usando le intestazioni LSB in ogni script in `/etc/init.d/`, oltre all'avvio concorrente parallelo attraverso l'utilizzo di `startpar` per velocizzare il processo di avvio.

Gli script in ognuna delle directory `/etc/rcN.d/` sono solo collegamenti simbolici agli script in `/etc/init.d/`. Comunque, i nomi dei file in ognuna delle directory `/etc/rcN.d/` sono selezionati per indicare il modo in cui gli script in `/etc/init.d/` vengono eseguiti. Specificatamente, prima di entrare in qualsiasi runlevel, sono eseguiti tutti gli script che iniziano con "K"; questi script uccidono i servizi. Poi vengono eseguiti tutti gli script che iniziano con "S"; questi script avviano i servizi. Il numero a due cifre che segue la "K" o la "S" indica l'ordine in cui lo script sarà eseguito. Gli script con un numero minore sono eseguiti prima.

Questo approccio funziona perché tutti gli script in `/etc/init.d/` accettano un argomento che può essere "start", "stop", "reload", "restart" o "force-reload" e svolgeranno poi il compito indicato dall'argomento. Questi script possono essere usati anche dopo che un sistema è stato avviato, per controllare vari processi.

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

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

invia al demone `sendmail` il segnale di rileggere il suo file di configurazione.

Notare che `invoke-rc.d` non dovrebbe essere usato per chiamare gli script `/etc/init.d/`, ma dovrebbe essere usato invece `service`.

11.6 Quale altre funzionalità sono fornite per personalizzare il processo di avvio a parte `rc.local`?

Lo script `rc.local` viene eseguito alla fine di ogni runlevel multiutente. In Debian è configurato per non fare nulla. Ciò fornisce personalizzazione del processo di avvio ma potrebbe non essere sufficiente in tutte le situazioni.

Si supponga che un sistema necessiti di eseguire lo script `pippo` all'avvio o all'ingresso ad un particolare runlevel (System V). Allora l'amministratore di sistema dovrebbe:

¹Al momento della stesura di questo testo il rilascio unstable per Debian GNU/Linux sta sostituendo `sysvinit` con `systemd`, un gestore di sistema e servizi per Linux. Per maggiori informazioni vedere <https://wiki.debian.org/systemd>.

- Mettere lo script `pippo` nella directory `/etc/init.d/`.
- Eseguire il comando `Debian update-rc.d` con gli argomenti appropriati, per specificare quali runlevel devono avviare il servizio e quali devono fermarlo.
- Considerare l'opportunità di riavviare il sistema per controllare che il servizio venga avviato correttamente (nell'ipotesi che si sia impostato il suo avvio nel runlevel predefinito). Altrimenti avviarlo manualmente eseguendo `"/etc/init.d/pippo start"`.

Si può, per esempio, fare in modo che lo script `pippo` venga eseguito all'avvio, mettendolo in `/etc/init.d/` ed eseguendo `update-rc.d pippo defaults 19`. L'argomento «defaults» fa riferimento ai runlevel predefiniti, il che significa (almeno in assenza di blocchi di commento LSB che specifichino diversamente) che il servizio viene avviato nei runlevel dal 2 al 5 e che viene fermato nei runlevel 0, 1 e 6. (Quasiassi direttiva LSB `Default-Start` e `Default-Stop` in `pippo` ha la precedenza quando si usa la versione `sysv-rc` di `update-rc.d`, ma viene ignorata dalla variante `file-rc` (v0.8.10 e successive) di `update-rc.d`.) L'argomento «19» assicura che `pippo` venga chiamato dopo che siano stati completati tutti gli script il cui numero è minore di 19 e prima di tutti gli script con numero uguale o maggiore di 20.

11.7 Come si comporta il sistema di manutenzione dei pacchetti nel caso di pacchetti che contengono file di configurazione per altri pacchetti?

Alcuni utenti desiderano creare, per esempio, un nuovo server installando un gruppo di pacchetti Debian ed un pacchetto generato localmente che consiste di file di configurazione. Questa non è generalmente una buona idea, perché `dpkg` non saprà nulla di quei file di configurazione se sono in un pacchetto differente, e potrebbe scrivere configurazioni in conflitto quando uno dei pacchetti del “gruppo” iniziale viene aggiornato.

Piuttosto, si crei un pacchetto locale che modifichi i file di configurazione del “gruppo” di pacchetti Debian che interessano. Allora `dpkg` e il resto del sistema di gestione dei pacchetti vedono che i file sono stati modificati dall'“amministratore di sistema” locale e non cercheranno di sovrascriverli quando quei pacchetti verranno aggiornati.

11.8 Come sovrascrivere un file installato da un pacchetto in modo che ne venga usata una versione differente?

Si supponga che l'amministratore o un utente locale desideri usare un programma “login-local” piuttosto del programma “login” fornito dal pacchetto `login` di Debian.

Non:

- Sovrascrivere `/bin/login` con `login-local`.

Il sistema di gestione dei pacchetti non saprà di questo cambiamento e semplicemente sovrascriverà il `/bin/login` personalizzato ogni volta che `login` (o qualsiasi altro pacchetto che fornisce `/bin/login`) verrà installato o aggiornato.

Invece,

- Eseguire:

```
dpkg-divert --divert /bin/login.debian /bin/login
```

in modo che tutte le future installazioni del pacchetto `login` di Debian scrivano invece il file `/bin/login` in `/bin/login.debian`.

- Poi eseguire:

```
cp login-local /bin/login
```

per spostare il proprio programma al suo posto.

Eseguire `dpkg-divert --list` per vedere quali deviazioni siano attualmente attive sul proprio sistema.

I dettagli sono forniti nella pagina di manuale `dpkg-divert(8)`.

11.9 Come si può far sì che un pacchetto generato localmente venga incluso nella lista dei pacchetti disponibili che il sistema di gestione dei pacchetti conosce?

Eseguire il comando:

```
dpkg--scanpackages BIN_DIR OVERRIDE_FILE [PREFISSOPERCORSO] > mio_Packages
```

dove:

- BIN-DIR è la directory dove sono contenuti i file di archivio Debian (che solitamente hanno estensione “.deb”).
- OVERRIDE_FILE è un file che viene modificato dai manutentori della distribuzione ed è solitamente situato, per i pacchetti Debian nella sezione “main” in `indices/override.main.gz` nell’archivio FTP Debian. Può essere ignorato per pacchetti locali.
- PREFISSOPERCORSO è una stringa *opzionale* che può essere posta prima del file `mio_Packages` prodotto.

Una volta che si è creato il file `mio_Packages`, si informi il sistema di gestione dei pacchetti usando il comando:

```
dpkg --merge-avail mio_Packages
```

Se si sta usando APT, si può anche aggiungere il repository locale al proprio file `sources.list` (5).

11.10 Ad alcuni utenti piace mawk, ad altri gawk; ad alcuni piace vim, ad altri elvis; ad alcuni piace trn, ad altri tin; come supporta Debian le diversità?

Ci sono diversi casi in cui due pacchetti forniscono due versioni differenti di un programma, che forniscono entrambe le stesse funzionalità fondamentali. Gli utenti possono preferire l’una rispetto all’altra per abitudine o perché l’interfaccia utente di un pacchetto è in qualche modo più piacevole di quella di un altro. Altri utenti sullo stesso sistema possono fare una scelta differente.

Debian usa un sistema di pacchetti “virtuali” per permettere agli amministratori di sistema di scegliere (o lasciare che gli utenti scelgano) i propri strumenti preferiti quando ce ne sono due o più che forniscono la stessa funzionalità di base pur continuando ancora a soddisfare i requisiti in termini di dipendenze dei pacchetti senza specificare un particolare pacchetto.

Per esempio, potrebbero esistere due differenti versioni di lettori di newsgroup su un sistema. Il pacchetto del server per newsgroup potrebbe “raccomandare” la presenza di *un qualche* lettore di newsgroup sul sistema, ma la scelta tra `tin` e `trn` è lasciata all’utente. Ciò è realizzato avendo entrambi i pacchetti `tin` e `trn` che forniscono il pacchetto virtuale `news-reader`. *Quale* programma venga chiamato è determinato da un collegamento che punta dal file con il nome del pacchetto virtuale `/etc/alternatives/news-reader` al file selezionato, per esempio `/usr/bin/trn`.

Un solo collegamento è insufficiente per supportare pienamente l’uso di un programma alternativo; normalmente devono essere selezionate anche le pagine di manuale e possibilmente anche altri file di supporto. Lo script Perl `update-alternatives` fornisce un mezzo per garantire che tutti i file associati con uno specifico pacchetto siano selezionati come i predefiniti di sistema.

Per esempio, per verificare quale eseguibile fornisce “x-window-manager”, eseguire:

```
update-alternatives --display x-window-manager
```

Se lo si desidera cambiare, eseguire:

```
update-alternatives --config x-window-manager
```

e seguire le istruzioni sullo schermo (sostanzialmente premere il numero vicino alla voce che si preferisce).

Se, per qualche ragione, un pacchetto non si registra da solo come un window manager (si segnali il bug se c’è un errore) o se si usa un window manager dalla directory `/usr/local`, le selezioni sullo schermo non conterranno la propria voce preferita. Si può aggiornare il collegamento attraverso opzioni da riga di comando, così:

```
update-alternatives --install /usr/bin/x-window-manager \
x-window-manager /usr/local/bin/wmaker-cvs 50
```

Il primo argomento dell'opzione '-install' è il collegamento simbolico che punta a /etc/alternatives/NOME, dove NOME è il secondo argomento. Il terzo argomento è il programma al quale /etc/alternatives/NOME dovrebbe puntare e il quarto argomento è la priorità (maggiore è il valore maggiore è la probabilità che l'alternativa sia scelta automaticamente).

Per rimuovere un'alternativa che si è aggiunta eseguire semplicemente:

```
update-alternatives --remove x-window-manager /usr/local/bin/wmaker-cvs
```


Capitolo 12

Ottenere supporto per Debian GNU/Linux

12.1 Quale altra documentazione esiste su e per un sistema Debian?

- Istruzioni per l'installazione per l'attuale rilascio: vedere <http://www.debian.org/releases/stable/installmanual>.
- La "Debian GNU/Linux Reference" copre molti aspetti dell'amministrazione di sistema attraverso esempi di comandi di shell. Vengono forniti tutorial di base, suggerimenti ed altre informazioni per molti argomenti diversi che spaziano dall'amministrazione di sistema alla programmazione.

La si può ottenere con il pacchetto `debian-reference` o all'indirizzo <http://www.debian.org/doc/user-manuals#quick-reference>.

- Il manuale "Debian Policy" documenta i requisiti delle linee guida della distribuzione, ovvero la struttura e il contenuto dell'archivio Debian, vari aspetti relativi alla progettazione del sistema operativo, ecc. Include anche i requisiti tecnici che ogni pacchetto deve soddisfare per essere incluso nella distribuzione e documenta gli aspetti tecnici di base dei pacchetti binari e sorgenti di Debian.

Lo si può ottenere con il pacchetto `debian-policy` o su <http://www.debian.org/doc/devel-manuals#policy>.

- La documentazione sviluppata dal progetto Debian Documentation Project. È disponibile su <http://www.debian.org/doc/> e include guide utente, guide per gli amministratori e guide di sicurezza relative al sistema operativo Debian GNU/Linux.
- La documentazione sui pacchetti Debian installati: la maggior parte dei pacchetti ha dei file che vengono estratti in `/usr/doc/PACCHETTO`.
- La documentazione sul progetto Linux. Il pacchetto Debian `doc-linux` installa tutte le più recenti versioni degli HOWTO e mini-HOWTO del Linux Documentation Project (<http://www.tldp.org/>).
- Le pagine "man" in stile Unix. La maggior parte dei comandi ha le pagine di manuale scritte nello stile dei file "man" originali di Unix. Per esempio, per vedere la pagina di manuale del comando "ls", eseguire `man ls`. Eseguire `man man` per maggiori informazioni su come trovare e visualizzare le pagine di manuale.

I nuovi utenti Debian dovrebbero notare che le pagine "man" di molti comandi generici di sistema non sono disponibili finché non si installano i seguenti pacchetti:

- `man-db`, che contiene il programma `man` stesso ed altri programmi per manipolare le pagine di manuale.
- `manpages`, che contiene le pagine di manuale di sistema. (Si veda 'Come supporta Debian le lingue diverse dall'inglese?' a pagina 19.)

- Le pagine "info" in stile GNU. La documentazione utente per molti comandi, in particolare per gli strumenti GNU, non è disponibile in forma di pagine di manuale, ma in file "info" che possono essere letti dallo strumento GNU `info`, eseguendo `M-x info` all'interno di GNU Emacs o con qualche altro visualizzatore di pagine Info.

Its main advantage over the original 'man' pages is that it is a hypertext system. It does *not* require the WWW, however; `info` can be run from a plain text console. It was designed by Richard Stallman and preceded the WWW.

Si noti che si può accedere a molta documentazione presente nel proprio sistema utilizzando un browser WWW, attraverso i comandi "dwww", "dhelp" o "doccentral", che si trovano nei rispettivi pacchetti oppure usando "yelp".

12.2 Ci sono risorse on-line dove discutere su Debian?

Sì. Infatti, il metodo principale con cui Debian fornisce supporto ai propri utenti è via posta elettronica. Verranno qui forniti alcuni dettagli e nominate alcune altre utili fonti di informazioni. Ulteriori risorse sono elencate nella pagina web Debian sul supporto (<http://www.debian.org/support>).

12.2.1 Mailing-list

Ci sono molte mailing-list relative a Debian (<http://www.debian.org/MailingLists/>)

Su di un sistema con il pacchetto `doc-debian` installato c'è un elenco completo delle mailing-list in `/usr/share/doc/debian/mailling-lists.txt`.

Le mailing-list Debian sono denominate seguendo il modello *debian-argomento-lista*. Esempi sono *debian-announce*, *debian-user*, *debian-news*. Per iscriversi a una qualsiasi delle liste *debian-argomento-lista*, inviare un messaggio di posta a *debian-argomento-lista-request@lists.debian.org* con la parola "subscribe" nell'intestazione Subject. Ci si assicuri di aver aggiunto *-request* all'indirizzo di posta elettronica quando si utilizza questo metodo per iscriversi o disiscriversi. In caso contrario il messaggio di posta va alla lista stessa, il che potrebbe essere imbarazzante o fastidioso, a seconda del punto di vista.

Ci si può iscrivere alle mailing-list utilizzando il modulo WWW (<http://www.debian.org/MailingLists/subscribe>). Ci si può anche disiscrivere utilizzando un modulo WWW (<http://www.debian.org/MailingLists/unsubscribe>).

L'indirizzo di posta elettronica dell'amministratore delle mailing-list è `<listmaster@lists.debian.org>`, nel caso si abbia qualsiasi problema.

Le mailing-list sono luoghi di discussione pubblici. Tutti i messaggi inviati alle liste vengono anche copiati nell'archivio pubblico dove tutti (compreso chi non è iscritto) possono sfogliarli o fare ricerche. Assicurarsi di non inviare mai alle liste materiale confidenziale o per il quale non si hanno i diritti. Sono incluse cose come gli indirizzi di posta elettronica; è degno di attenzione particolare il fatto che le fonti di spam sono note per abusare degli indirizzi di posta elettronica inviati alle mailing-list Debian. Si veda la politica sulla privacy delle mailing-list (<http://www.debian.org/MailingLists/#disclaimer>) per maggiori informazioni.

Gli archivi delle mailing-list Debian sono disponibili via WWW all'indirizzo <http://lists.debian.org/>.

Qual è il codice di condotta per le mailing-list?

Quando si usano le mailing-list Debian, seguire queste regole:

- Non inviare spam. Si veda la Politica Debian per la pubblicità sulle mailing-list (<http://www.debian.org/MailingLists/#ads>).
- Evitare i flame; non è educato. Le persone che sviluppano Debian sono tutti volontari, che donano il proprio tempo, energia e denaro nel tentativo di portare avanti il progetto Debian insieme.
- Non usare un linguaggio volgare; inoltre alcune persone ricevono le liste tramite pacchetti radio, dove imprecare è illegale.
- Assicurarsi di utilizzare la lista corretta. *Mai* inviare la propria richiesta di (dis)iscrizione alla mailing-list stessa.¹
- Vedere la sezione 'Come si segnala un bug in Debian?' a fronte per indicazioni sulla segnalazione di bug.

12.2.2 Forum web

`debianHELP` (<http://debianhelp.org/>) e `Debian User Forums` (<http://forums.debian.net/>) sono forum web in cui si possono porre domande su Debian e ottenere risposte da altri utenti. (Non fanno ufficialmente parte del progetto Debian.)

12.2.3 Wiki

Le soluzioni a problemi comuni, HOWTO, guide, suggerimenti e altra documentazione possono essere trovati nel Wiki Debian (<http://wiki.debian.org/>) in costante evoluzione.

¹Utilizzare l'indirizzo `debian-argomento-lista-REQUEST@lists.debian.org` per quello.

12.2.4 Manutentori

Gli utenti possono rivolgere domande ai manutentori dei singoli pacchetti usando la posta elettronica. Per raggiungere un manutentore di un pacchetto chiamato *xyz*, si invii un messaggio a *xyz@packages.debian.org*.

12.2.5 Gruppi di discussione (newsgroup) Usenet

Gli utenti dovrebbero rivolgere le domande non specifiche per Debian in uno dei gruppi USENET Linux, che si chiamano *comp.os.linux.** o *linux.** [NdT: in italiano esistono i gruppi *it.comp.os.linux.**]. Ci sono diversi elenchi di gruppi di discussione Usenet Linux ed altre risorse correlate sul WWW, ad esempio sui siti Linux Online (<http://www.linux.org/docs/usenet.html>) e LinuxJournal (<http://www.linuxjournal.com/helpdesk.php>).

12.3 Esiste un modo rapido per cercare informazioni su Debian GNU/Linux?

Ci sono svariati motori di ricerca che forniscono documentazione relativa a Debian:

- sito WWW di ricerca Debian (<http://search.debian.org/>).
- Google Groups (<http://groups.google.com/>): un motore di ricerca per gruppi di discussione.
For example, to find out what experiences people have had with finding drivers for NVIDIA graphic cards under Debian, try searching the phrase *NVIDIA Linux driver*. This will show you all the posts that contain these strings, i.e. those where people discussed these topics. If you add *Debian* to those search strings, you'll also get the posts specifically related to Debian.
- Any of the common web spidering engines, such as DuckDuckGo (<https://duckduckgo.com/>) or Google (<http://www.google.com/>), as long as you use the right search terms.
For example, searching on the string "evince" gives a more detailed explanation of this package than the brief description field in its control file.

12.4 Ci sono registri dei bug conosciuti?

Le segnalazioni dei problemi irrisolti (e su quelli chiusi) sono disponibili pubblicamente: Debian ha promesso di fare così dichiarando nel Contratto Sociale Debian (http://www.debian.org/social_contract) "Noi non nasconderemo i problemi".

La distribuzione Debian GNU/Linux ha un sistema di tracciamento dei bug (BTS, bug tracking system) che registra i dettagli dei bug riportati da utenti e sviluppatori. Ad ogni bug viene assegnato un numero e viene archiviato. Una volta che è stato risolto, viene segnato come tale.

Copie di queste informazioni sono disponibili su <http://www.debian.org/Bugs/>.

Un server di posta fornisce accesso al database del sistema di tracciamento dei bug via posta elettronica. Per ottenere le istruzioni si invii un messaggio a request@bugs.debian.org con "help" nel corpo del messaggio.

12.5 Come si segnala un bug in Debian?

Se si è trovato un bug in Debian si leggano le istruzioni per la segnalazione di un bug. Queste istruzioni possono essere ottenute in uno di vari modi.

- Dal WWW. Una copia delle istruzioni è mostrata in <http://www.debian.org/Bugs/Reporting>.
- Su qualsiasi sistema Debian con installato il pacchetto *doc-debian*. Le istruzioni sono nel file `/usr/share/doc/debian/bug-reporting.txt`.
- Via FTP anonimo. I siti mirror Debian contengono le istruzioni nel file `doc/bug-reporting.txt`.

Si può usare il pacchetto `reportbug` che guida l'utente attraverso il processo di segnalazione ed invia il messaggio all'indirizzo corretto, aggiungendo automaticamente alcuni dettagli aggiuntivi sul sistema. Mostra anche un elenco di bug già segnalati per il pacchetto interessato dalla segnalazione in corso, in modo che, nel caso che il bug sia già stato segnalato in precedenza, si possono aggiungere ulteriori informazioni alla segnalazione esistente.

Ci si aspetta di ricevere un messaggio automatico di conferma della propria segnalazione. A questa verrà anche assegnato un numero di tracciamento bug, verrà immessa nel registro dei bug ed inoltrata alla mailing-list `debian-bugs-dist`.

Capitolo 13

Contribuire al Progetto Debian

Donazioni (<http://www.debian.org/donations>) di tempo (sviluppare nuovi pacchetti, mantenere pacchetti esistenti o fornire supporto agli utenti), risorse (fare da mirror agli archivi FTP e WWW) e denaro (pagare per nuove piattaforme di test come pure hardware per gli archivi) possono aiutare il progetto. Vedere anche Come posso aiutare Debian? (<http://www.debian.org/intro/help>).

13.1 How can I become a Debian member/Debian developer?

The development of Debian is open to all, and new users with the right skills and/or the willingness to learn are needed to maintain existing packages which have been “orphaned” by their previous maintainers, to develop new packages, to write documentation, to do translation work, to help with the Debian website, to provide user support, etc.

The description of becoming a Debian member can be found at the New Member’s Corner (<http://www.debian.org/devel/join/newmaint>) at the Debian web site.

13.2 Come si possono fornire risorse al progetto Debian?

Dato che il progetto mira a creare un corpo esaustivo di software accessibile rapidamente e facilmente da tutto il mondo, c’è necessità di mirror. È desiderabile, ma non assolutamente necessario, fare il mirror di tutto l’archivio. Si visiti la pagina Dimensione del mirror Debian (<http://www.debian.org/mirror/size>) per informazioni sui requisiti di spazio su disco.

La maggior parte del lavoro di mirror è realizzata interamente in modo automatico da degli script senza alcuna interazione. Comunque, accadono occasionali errori o modifiche di sistema che richiedono l’intervento umano.

Se si possiedono una connessione ad Internet ad alta velocità, i mezzi per fare il mirror di tutta o di parte della distribuzione e si è disposti a dedicare tempo (o a trovare qualcuno) per poter fornire una regolare manutenzione al sistema, allora contattare `<debian-admin@lists.debian.org>`.

13.3 Come si può contribuire finanziariamente al progetto Debian?

Le donazioni degli sponsor permettono a Debian, tra le altre cose, di avere macchine, oltre ad altro hardware, organizzare conferenze e sprint di sviluppo. Per maggiori informazioni visitare Debian Donations (<http://www.debian.org/donations>). La pagina elenca anche i diversi metodi che possono essere usati per donare.

Si possono fare donazioni individuali a organizzazioni che sono vitali per lo sviluppo del progetto Debian. L’organizzazione principale è Software in the Public Interest, negli Stati Uniti, ma ce ne sono altre.

13.3.1 Software in the Public Interest

Software in the Public Interest (SPI) è un’organizzazione IRS 501(c)(3) senza fini di lucro con base negli Stati Uniti. Lo scopo dell’organizzazione è quello di sviluppare e distribuire software libero.

Incoraggia i programmatori ad usare la licenza GNU General Public License o altre licenze che permettano la redistribuzione e l'uso liberi del software e gli sviluppatori hardware a distribuire documentazione che permetta la scrittura di driver di dispositivo per il loro prodotto.

SPI agisce come sponsor fiscale per molti progetti liberi e open source. Il progetto Debian è un progetto associato sin dalla creazione dell'organizzazione.

SPI può essere raggiunta su: <http://www.spi-inc.org/>.

13.3.2 Altre organizzazioni

There are a number of organizations created in different countries that hold assets in trust for Debian. The donations page (<http://www.debian.org/donations>) lists the trusted organizations individuals can donate to. At the time of this writing there are three of them: Verein zur Forderung Freier Informationen & Software (<http://www.ffis.de/>) (in Germany), the Debian France Association (<https://france.debian.net/>) (in France), and debian.ch (<http://debian.ch/>) (Switzerland and the Principality of Liechtenstein). Additional affiliate organizations in other countries are listed in Organizations (<https://wiki.debian.org/Teams/Auditor/Organizations>) page in the Debian Wiki.

Capitolo 14

Ridistribuire Debian GNU/Linux in un prodotto commerciale

14.1 Si possono creare e vendere CD Debian?

Certamente. Non è necessario il permesso per distribuire qualcosa che abbiamo *rilasciato*, quindi si può masterizzare il proprio CD appena finisce il beta-test. Non si deve pagare niente. Ovvio che tutti i produttori di CD devono rispettare le licenze dei programmi in Debian. Per esempio, molti dei programmi sono sotto licenza GPL, che richiede che si redistribuisca il loro codice sorgente.

Inoltre, viene pubblicato un elenco dei produttori di CD che donano denaro, software e tempo al progetto Debian, e gli utenti sono incoraggiati a comperare dai produttori che donano, quindi fare donazioni è una buona pubblicità.

14.2 Si può impacchettare Debian con software non libero?

Sì. Sebbene tutti i componenti principali di Debian siano software libero, è fornita una directory non-free per programmi che non sono liberamente redistribuibili.

I produttori di CD *potrebbero* essere in grado di distribuire i programmi che sono stati collocati in quella directory, a seconda dei termini delle licenze o dei loro accordi privati con gli autori di quei pacchetti software. I produttori di CD possono inoltre distribuire il software non-libero che ricevono da altre fonti sullo stesso CD. Non è niente di nuovo: software libero e commerciale sono ora distribuiti sullo stesso CD da molti produttori. Ovvio che incoraggiamo sempre gli autori del software a rilasciare i programmi che scrivono come software libero.

14.3 Sto creando una speciale distribuzione Linux per un “mercato verticale”. Posso usare Debian GNU/Linux come nucleo di un sistema Linux e aggiungere le mie applicazioni su di esso?

Yes. Debian-derived distributions are being created both in close cooperation with the Debian project itself and by external parties. One can use the Debian Pure Blends (<https://www.debian.org/blends/>) framework to work together with Debian; DebianEdu/Skolelinux (<https://wiki.debian.org/DebianEdu/>) is one such project.

There are several other Debian-derived distributions already on the market, such as grml, LMDE (Linux Mint Debian Edition), Knoppix and Ubuntu, that are targeted at a different kind of audience than the original Debian GNU/Linux is, but use most of our components in their product.

Debian fornisce inoltre un meccanismo che permette agli sviluppatori e agli amministratori di sistema di installare versioni locali di file specifici in modo che non vengano sovrascritti quando altri pacchetti vengono aggiornati. Questo è discusso meglio nella domanda su ‘Come sovrascrivere un file installato da un pacchetto in modo che ne venga usata una versione differente?’ a pagina 49.

14.4 È possibile mettere un proprio programma commerciale in un “pacchetto” Debian cosicché si possa installare senza fatica su ogni sistema Debian?

Certamente. Lo strumento per i pacchetti è software libero; i pacchetti possono essere o meno software libero, li può installare tutti.

Capitolo 15

Cambiamenti attesi nel prossimo rilascio principale di Debian

Con ogni nuovo rilascio il progetto Debian cerca di concentrarsi su un insieme di argomenti. Questi sono noti come “Obiettivi di rilascio” e sono tutti descritti in <https://wiki.debian.org/ReleaseGoals/>. Notare che le sezioni seguenti potrebbero non essere completamente aggiornate; fare riferimento al Wiki per maggiori informazioni e per lo stato aggiornato di questi obiettivi.

15.1 Rafforzare il sistema

È un obiettivo del progetto Debian assicurare che qualsiasi sistema installato sia rafforzato e sicuro contro gli attacchi. Ci sono diversi modi per ottenerlo che includono:

- Migliorare la sicurezza dei programmi compilandoli con Security Hardening Build Flags (<https://wiki.debian.org/ReleaseGoals/SecurityHardeningBuildFlags>) (opzioni di compilazione di rafforzamento della sicurezza) allo scopo di abilitare varie protezioni contro problemi di sicurezza noti.
- Migliorare la configurazione predefinita di sistema per renderlo meno vulnerabile ad attacchi (sia in locale sia da remoto).
- Abilitare funzionalità di sicurezza fornite da versioni nuove del kernel.

All of these are done in an ongoing basis. For the first item, a set of security hardening build flags that try to prevent known attacks such as stack smashing, predictable locations of values in memory, etc. is used. The target is to cover at least all packages that are part of the basic installation as well as packages that had to be updated through a Security Advisory since 2006. As of this writing, around 400 packages have been modified since this effort was first started. All the issues are tracked in the BTS (<https://bugs.debian.org/cgi-bin/pkgreport.cgi?tag=goal-hardening;users=hardening-discuss@lists.alioth.debian.org>).

15.2 Supporto esteso per gli utenti non-inglesi

Debian ha già un ottimo supporto per utenti non-inglesi, si veda ‘Come supporta Debian le lingue diverse dall’inglese?’ a pagina 19.

Ci si augura di trovare persone che forniscano supporto per ancora più lingue e che traducano programmi e documenti. Molti programmi e documenti specifici di Debian supportano già l’internazionalizzazione, così abbiamo bisogno di traduttori dei cataloghi dei messaggi. Tuttavia rimangono ancora programmi che devono essere correttamente internazionalizzati.

Il GNU Translation Project <ftp://ftp.gnu.org/pub/gnu/ABOUT-NLS> lavora sull’internazionalizzazione dei programmi GNU e diversi progetti, come gli ambienti desktop GNOME e KDE, hanno dei propri gruppi di traduzione. L’obiettivo di Debian non è quello di sostituire o ripetere il lavoro fatto da questi progetti; in verità Debian trae beneficio dal lavoro fatto dai traduttori in questi progetti. Tuttavia ci sono ancora molti programmi che non rientrano nell’ambito di questi progetti e che sono tradotti all’interno di Debian.

I precedenti rilasci di Debian si sono concentrati su aspetti come:

- Supporto i18n in tutti i pacchetti che usano debconf: i pacchetti che usano la gestione Debian della configurazione devono permettere la traduzione di tutti i messaggi presentati all'utente durante la configurazione del pacchetto.
- Supporto i18n per le descrizioni dei pacchetti: aggiornare le interfacce alla gestione dei pacchetti in modo che usino le descrizioni tradotte dei pacchetti.
- debian/changelog e debian/control UTF-8. In questo modo è possibile, per esempio, stampare correttamente nei changelog i nomi delle persone provenienti dai paesi asiatici.
- Supporto i18n nell'installatore Debian incluso il pieno supporto per alcune lingue che richiedono l'uso dell'interfaccia grafica.

15.3 Miglioramenti nell'installatore Debian

È stato fatto molto lavoro sull'installatore Debian che ha portato a grandi miglioramenti. Ne citeremo qui solo due.

Avvio dell'installatore da Microsoft Windows: è ora possibile avviare l'installatore direttamente da Microsoft Windows senza bisogno di modificare le impostazioni del BIOS. All'inserimento di un CD-ROM, DVD-ROM o pennina USB, un programma eseguito automaticamente verrà avviato, offrendo una procedura passo-passo per avviare l'installatore Debian.

15.4 Più architetture

Complete Debian system on other architectures. Notice that even though some architectures are dropped for a given release, there still might be a way to install and upgrade using the latest `sid`.

15.5 Più kernel

In aggiunta a Debian GNU/Hurd, Debian sta facendo il port anche per i kernel BSD, in particolare per FreeBSD (<http://www.debian.org/ports/kfreebsd-gnu/>). Questo port gira sia su macchine AMD64 ("kfreebsd-amd64") sia su Intel tradizionali ("kfreebsd-i386").

Capitolo 16

Informazioni generali su questa FAQ

16.1 Autori

La prima edizione di questa FAQ è stata fatta ed era mantenuta da J.H.M. Dassen (Ray) e Chuck Stickelman. Gli autori della Debian GNU/Linux FAQ riscritta sono Susan G. Kleinmann e Sven Rudolph. Dopo di loro, la FAQ è stata mantenuta da Santiago Vila e, in seguito, da Josip Rodin. L'attuale manutentore è Javier Fernandez-Sanguino.

Parti delle informazioni provengono da:

- L'annuncio del rilascio Debian-1.1, di Bruce Perens (<http://www.perens.com/>);
- le Linux FAQ di Ian Jackson (<http://www.chiark.greenend.org.uk/~ijackson/>);
- gli archivi delle mailing-list Debian (<http://lists.debian.org/>);
- il manuale per programmatori di dpkg e il manuale Debian Policy (si veda 'Quale altra documentazione esiste su e per un sistema Debian?' a pagina 53);
- molti sviluppatori, volontari e beta tester e
- la memoria traballante dei suoi autori :-)
- e inoltre la Choosing a Debian distribution FAQ (http://KamarajuKusumanchi.github.io/choosing_debian_distribution/choosing_debian_distribution.html), che Kamaraju Kusumanchi ha gentilmente rilasciato sotto GPL in modo che potesse essere inclusa come un nuovo capitolo (si veda 'Scegliere una distribuzione Debian' a pagina 7).

Gli autori desiderano ringraziare tutti coloro che hanno aiutato a rendere possibile questo documento.

Non esistono garanzie. Tutti i marchi registrati appartengono ai rispettivi proprietari.

16.2 Feedback

Commenti ed aggiunte a questo documento sono sempre i benvenuti. Mandare un messaggio di posta elettronica a [<doc-debian@packages.debian.org>](mailto:doc-debian@packages.debian.org) o inviare una segnalazione di bug di tipo wishlist sul pacchetto `debian-faq` (<http://bugs.debian.org/debian-faq>).

16.3 Reperibilità

La versione più recente di questo documento può essere letta sulle pagine WWW di Debian all'indirizzo <http://www.debian.org/doc/FAQ/>.

È anche disponibile per essere scaricata nei formati testo puro, HTML, PostScript e PDF all'indirizzo <http://www.debian.org/doc/user-manuals#faq>. Inoltre, ci sono lì anche diverse traduzioni.

Questo documento è disponibile nel pacchetto `debian-faq`. Traduzioni sono disponibili in `debian-faq-de`, `debian-faq-fr` e altri pacchetti.

I file SGML originali usati per creare questo documento sono inoltre disponibili nel pacchetto sorgente `debian-faq` o nel SVN agli indirizzi: <svn://svn.debian.org/svn/ddp/manuals/trunk/debian-faq> e <http://svn.debian.org/viewsvn/ddp/manuals/trunk/debian-faq/>.

16.4 Formato del documento

Questo documento è stato scritto usando il DTD SGML DebianDoc (riscritto da LinuxDoc SGML). I sistemi SGML DebianDoc permettono di creare file in una varietà di formati da un unico sorgente, per esempio questo documento può essere visto come HTML, testo puro, TeX DVI, PostScript, PDF e GNU info.

Le utilità di conversione per l'SGML DebianDoc sono disponibili nel pacchetto Debian `debiandoc-sgml`.