

The Debian GNU/Linux FAQ

2 ottobre 2019

The Debian GNU/Linux FAQ

Published 12 August 2019

Copyright © 1996-2019 Software in the Public Interest

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?	3
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?	4
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?	6
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?	9
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?	10
3.1.9	Cosa succede quando viene fatto un nuovo rilascio?	10
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	Al momento sto usando stable. Posso passare a testing o unstable? Se sì, come?	11
3.1.12	Attualmente sto usando testing (bullseye). 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?	12
3.2	E per quanto riguarda Knoppix, Linux Mint Debian Edition, Ubuntu e altre?	12
3.2.1	So che Knoppix/Linux Mint Debian Edition/Ubuntu/... è basata su Debian. Perciò dopo l'installazione sull'hard disk, posso usare con essa gli strumenti «apt» per i pacchetti?	12
3.2.2	Ho installato Knoppix/Linux Mint Debian Edition/Ubuntu/... sul mio hard disk. Ora ho un problema. Cosa devo fare?	12
3.2.3	Sto usando Knoppix/LMDE/Ubuntu/... e ora voglio usare Debian. Come migro a Debian?	13

4	Problemi di compatibilità	15
4.1	Su quali architetture o sistemi hardware funziona Debian?	15
4.2	Quali kernel può eseguire Debian GNU/Linux?	16
4.3	Quanto è compatibile Debian con le altre distribuzioni Linux?	16
4.4	Quanto è compatibile Debian con altri sistemi Unix a livello di codice sorgente?	16
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?	17
4.6	Come installare un programma non-Debian?	17
5	Software disponibile nel sistema Debian	19
5.1	Quali tipi di applicazioni e software di sviluppo sono disponibili per Debian GNU/Linux?	19
5.2	Chi ha scritto tutto questo software?	19
5.3	Come si può ottenere una lista aggiornata dei programmi che sono stati impacchettati per Debian?	20
5.4	Come si può installare un ambiente di sviluppo per compilare pacchetti?	20
5.5	Cosa manca da Debian GNU/Linux?	20
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?	20
5.7	Debian supporta Java? (E come?)	20
5.8	Come si può verificare di stare usando un sistema Debian e che versione è?	21
5.9	Come supporta Debian le lingue diverse dall'inglese?	21
5.10	Dove è ezmlm/djbdns/qmail?	22
5.11	Dove è un riproduttore per Flash (SWF)?	22
5.12	Dove è Google Earth?	22
5.13	Dove è il software VoIP?	22
5.14	Ho una scheda di rete wireless che non funziona con Linux. Cosa fare?	22
6	Gli archivi FTP Debian	23
6.1	Quante distribuzioni di Debian ci sono?	23
6.2	Cosa sono tutti quei nomi come etch, lenny, ecc.?	23
6.2.1	Quali altri nomi in codice sono stati usati in passato?	23
6.2.2	Da dove derivano questi nomi in codice?	23
6.3	Cosa dire a proposito di "sid"?	24
6.4	Cosa contiene la directory stable?	24
6.5	Cosa contiene la distribuzione testing?	25
6.5.1	Cosa dire di "testing"? Come viene "congelata"?	25
6.6	Cosa contiene la distribuzione unstable?	25
6.7	Cosa sono tutte quelle directory negli archivi FTP Debian?	26
6.8	Cosa sono tutte quelle directory dentro a dists/stable/main?	26
6.9	Dove è il codice sorgente?	26
6.10	Cosa c'è nella directory pool?	27
6.11	Cos'è "incoming"?	27
6.12	Come impostare un proprio repository usabile con apt?	27
7	Fondamenti sul sistema di gestione dei pacchetti Debian	29
7.1	Cosa è un pacchetto Debian?	29
7.2	Qual è il formato di un pacchetto binario Debian?	30
7.3	Perché i nomi dei pacchetti Debian sono così lunghi?	30
7.4	Cos'è un file di controllo Debian?	30
7.5	Cos'è un conf file Debian?	31
7.6	Cosa sono gli script preinst, postinst, prerm e postrm di Debian?	31
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> ?	32
7.8	Cos'è un pacchetto virtuale?	32
7.9	Cosa significa dire che un pacchetto <i>Dipende</i> da, <i>Raccomanda</i> , <i>Suggerisce</i> , <i>Va in conflitto</i> con, <i>Sostituisce</i> , <i>Rompe</i> o <i>Fornisce</i> un altro pacchetto?	33
7.10	Cosa significa Pre-Depends?	33
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?	34
7.12	Come si blocca un pacchetto?	34

7.13	Come si installa un pacchetto sorgente?	35
7.14	Come si compilano pacchetti binari da un pacchetto sorgente?	35
7.15	Come ci si creano pacchetti Debian da sé?	36
8	Gli strumenti di gestione dei pacchetti Debian	37
8.1	Quali programmi fornisce Debian per gestire i suoi pacchetti?	37
8.1.1	dpkg	37
8.1.2	APT	38
8.1.3	aptitude	39
8.1.4	synaptic	40
8.1.5	tasksel	40
8.1.6	Altri strumenti per la gestione dei pacchetti	40
8.1.6.1	dpkg-deb	40
8.2	Debian afferma di essere in grado di aggiornare un programma in esecuzione; come viene fatto?	41
8.3	Come si può sapere quali pacchetti sono già installati su un sistema Debian?	41
8.4	Come mostrare i file di un pacchetto installato?	41
8.5	Come posso scoprire quale pacchetto ha prodotto un particolare file?	41
8.6	Perché «pippo-data» non viene rimosso quando viene disinstallato «pippo»? Come ci si può assicurare che vengano eliminati i vecchi pacchetti di librerie non utilizzati?	42
9	Mantenere il sistema Debian aggiornato	43
9.1	Come si può mantenere il proprio sistema Debian aggiornato?	43
9.1.1	aptitude	43
9.1.2	apt-get e apt-cdrom	44
9.1.3	mirror	44
9.2	È necessario entrare in modalità singolo-utente per aggiornare un pacchetto?	44
9.3	È necessario tenere tutti questi file di archivio .deb sul disco?	45
9.4	Come si può tenere un registro dei pacchetti che sono stati aggiunti al sistema? Vorrei sapere quali aggiornamenti e rimozioni di pacchetti sono stati fatti!	45
9.5	È possibile aggiornare automaticamente il sistema?	45
9.6	Se si hanno svariate macchine come è possibile scaricare gli aggiornamenti una volta sola?	45
10	Debian e il kernel	47
10.1	Si può installare e compilare un kernel senza alcun adattamento specifico per Debian?	47
10.2	Quali strumenti fornisce Debian per costruire kernel personalizzati?	47
10.3	Quali strumenti speciali fornisce Debian per lavorare con i moduli?	47
10.4	Si può disinstallare in sicurezza un vecchio pacchetto kernel e, se sì, come?	47
10.5	Come si possono ottenere ulteriori informazioni sui pacchetti Linux per Debian?	48
11	Personalizzare il proprio sistema Debian GNU/Linux	49
11.1	Come ci si può assicurare che tutti i programmi usino lo stesso formato per la carta?	49
11.2	Come si può fornire accesso alle periferiche hardware senza compromettere la sicurezza?	49
11.3	Come caricare un tipo di carattere per la console all'avvio nella maniera Debian?	49
11.4	Come si possono configurare le impostazioni predefinite per l'applicazione di un programma X11?	49
11.5	Come si avvia un sistema Debian?	50
11.6	Cosa ne è dell'init System V tradizionale in Debian?	50
11.7	Esistono ancora altri modi di avviare un sistema Debian?	51
11.8	Come si comporta il sistema di manutenzione dei pacchetti nel caso di pacchetti che contengono file di configurazione per altri pacchetti?	51
11.9	Come sovrascrivere un file installato da un pacchetto in modo che ne venga usata una versione differente?	52
11.10	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?	52
11.11	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?	53

12 Ottenere supporto per Debian GNU/Linux	55
12.1 Quale altra documentazione esiste su e per un sistema Debian?	55
12.2 Ci sono risorse on-line dove discutere su Debian?	56
12.2.1 Mailing-list	56
12.2.1.1 Qual è il codice di condotta per le mailing-list?	56
12.2.2 Forum web	57
12.2.3 Wiki	57
12.2.4 Manutentori	57
12.2.5 Gruppi di discussione (newsgroup) Usenet	57
12.3 Esiste un modo rapido per cercare informazioni su Debian GNU/Linux?	57
12.4 Ci sono registri dei bug conosciuti?	57
12.5 Come si segnala un bug in Debian?	58
13 Contribuire al Progetto Debian	59
13.1 Come si può diventare un membro o uno sviluppatore Debian?	59
13.2 Come si possono fornire risorse al progetto Debian?	59
13.3 Come si può contribuire finanziariamente al progetto Debian?	59
13.3.1 Software in the Public Interest	60
13.3.2 Altre organizzazioni	60
14 Ridistribuire Debian GNU/Linux in un prodotto commerciale	61
14.1 Si possono creare e vendere CD Debian?	61
14.2 Si può impacchettare Debian con software non libero?	61
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?	61
14.4 È possibile mettere un proprio programma commerciale in un "pacchetto" Debian cosicché si possa installare senza fatica su ogni sistema Debian?	62
15 Cambiamenti attesi nel prossimo rilascio principale di Debian	63
15.1 Rafforzare il sistema	63
15.2 Supporto esteso per gli utenti non-inglesi	63
15.3 Miglioramenti nell'installatore Debian	64
15.4 Più architetture	64
15.5 Più kernel	64
16 Informazioni generali su questa FAQ	65
16.1 Autori	65
16.2 Feedback	65
16.3 Reperibilità	65
16.4 Formato del documento	66
Indice analitico	67

Sommario

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

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 Sezione 12.1. Se anche ciò non aiuta, fare riferimento a Sezione 16.2.

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 58000 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](https://www.debian.org/distrib/ftplist) (<https://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 930 pacchetti software (nelle sezioni *non-free* e *contrib*), che sono distribuibili nei termini specifici indicati in ciascun pacchetto.
- **dinamica:** con circa 1343 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](https://www.debian.org/social_contract#guidelines) (https://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 Sezione [14.3](#) 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 Sezione [4.4](#).

Per maggiori informazioni su Linux, vedere [Cosa è Linux](https://www.linux.org/info/) (<https://www.linux.org/info/>) di [Linux Online](https://www.linux.org/) (<https://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 <https://www.gnu.org/software/hurd> per maggiori informazioni su GNU/Hurd in generale e <https://www.debian.org/ports/hurd/> per maggiori informazioni su Debian GNU/Hurd.

Un secondo sforzo è stato quello di fare il port per il kernel BSD. Persone stanno lavorando al kernel FreeBSD.

Si veda <https://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** (https://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)** (https://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 Capitolo 7).

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 fatto propria la causa della creazione di un sistema operativo libero, nello stesso spirito di Linux e di altro software libero.

Più di 1343 volontari che mantengono i pacchetti stanno lavorando su più di 58000 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 Capitolo 13.

Il Sistema Operativo Universale: Debian viene fornita con **più di 58000 pacchetti** (<https://packages.debian.org/stable/>) e gira su **10 architetture** (<https://www.debian.org/ports/>). Ciò è molto più di quello che è disponibile per le altre distribuzioni GNU/Linux. Si veda Sezione 5.1 per una panoramica del software fornito e Sezione 4.1 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 Sezione 12.4.

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** (https://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** (<https://www.gnu.org/>) ed in particolare da **Richard Stallman** (<https://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** (<https://www.gnu.org/software/hurd>). Debian sta lavorando con la FSF su questo sistema, chiamato **Debian GNU/Hurd** (<https://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](https://www.debian.org/releases/stable/installmanual) (<https://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 10, alias la distribuzione "stabile" o buster Questo è software stabile e ben testato, cambia se vengono incorporate soluzioni importanti a problemi di sicurezza e usabilità.

la distribuzione "testing", attualmente chiamata bullseye Qui è dove vengono messi i pacchetti che saranno rilasciati come prossima "stabile"; sono stati testati un po' in unstable ma possono non essere ancora completamente adatti ad essere rilasciati. Questa distribuzione è aggiornata più frequentemente della "stabile", 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 Sezione [6.1](#) 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](https://www.debian.org/security/faq) (<https://www.debian.org/security/faq>) o il [Debian Security Manual](https://www.debian.org/doc/manuals/securing-debian-howto/) (<https://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](https://www.debian.org/mirror/list) (<https://www.debian.org/mirror/list>).

Per maggiori informazioni su immagini di CD (e DVD) si faccia riferimento a [Debian GNU/Linux su CD](https://www.debian.org/CD) (<https://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](https://www.debian.org/releases/stable/installmanual) (<https://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 -> buster/  
/dists/stable -> buster/  
/dists/testing -> buster/  
/dists/unstable -> buster/
```

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 <https://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 <https://www.debian.org/CD/netinst/>.

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

Yes. Apart from CD or DVD, you can install Debian GNU/Linux by booting from USB memory stick, directly from hard disk, or using TFTP net booting. For installing on multiple computers it's possible to do fully automatic installations. NB: not all methods are supported by all computer architectures. Once the installer has booted, the rest of the system can be downloaded over the network, or installed from local media. See the [Debian GNU/Linux Installation Guide](https://www.debian.org/releases/stable/installmanual) (<https://www.debian.org/releases/stable/installmanual>) for more information.

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 Sezione 6.1.

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.
- Se si è un utente desktop con molta esperienza del sistema operativo e che non si preoccupa di dover affrontare un bug una volta ogni tanto, usare unstable. Ha tutto il software più recente e all'avanguardia ed i bug sono solitamente risolti prontamente.
- 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, <https://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.

Un'altra opzione sarebbe chiedere nella mailing list debian-user inviando un messaggio di posta elettronica a debian-user@lists.debian.org. È possibile inviare messaggi alla lista anche senza essere iscritti.

Gli archivi possono essere letti via <https://lists.debian.org/debian-user/>. Informazioni relative all'iscrizione alla lista sono reperibili all'indirizzo degli archivi. È caldamente consigliato porre le proprie domande sulla mailing-list piuttosto che su **IRC** (<https://www.debian.org/support>); i messaggi della mailing-list sono archiviati perciò la soluzione ai propri problemi potrà aiutare altri nella stessa situazione.

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.

I pacchetti in testing o unstable, d'altra parte, possono avere bug nascosti, falle di sicurezza, ecc. Inoltre alcuni pacchetti in testing e unstable potrebbero non funzionare come atteso. Di solito le persone che lavorano su una singola macchina desktop preferiscono avere l'insieme di pacchetti più recente e moderno. Unstable è la soluzione adatta per loro.

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.

In realtà, se si è esperti, si è disposti a spenderci del tempo, si è molto cauti e si sa quello che si sta facendo, allora potrebbe essere possibile passare da unstable a testing e successivamente a stable. Gli script di installazione non sono pensati per fare questo; perciò, nel farlo, i propri file di configurazione potrebbero andar perduti e...

3.1.5 Potete dirmi se installare stable, testing o unstable?

No. Questa è una questione piuttosto soggettiva. Non esiste una risposta perfetta dato che dipende dal software di cui si ha bisogno, la disponibilità ad affrontare i possibili malfunzionamenti e l'esperienza nell'amministrazione di sistema. Ecco alcuni suggerimenti:

- 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?

Le soluzioni dei bug e le migliorie introdotte nella distribuzione unstable passano in testing dopo un certo numero di giorni. Diciamo che la soglia è di 5 giorni. I pacchetti in unstable passano in testing solo quando non ci sono segnalazioni di bug critici per il rilascio che li riguardano. Se viene segnalato un bug critico per il rilascio riguardante un pacchetto in unstable, questo non passa a testing allo scadere dei 5 giorni.

L'idea è che, se il pacchetto ha dei problemi, questi vengono scoperti da coloro che usano unstable e vengono risolti prima che il pacchetto entri in testing. Ciò mantiene testing in uno stato usabile per la maggior parte del tempo. Nel suo insieme il concetto è impeccabile, a mio parere. Ma le cose non sono sempre così semplici. Prendiamo per esempio la situazione seguente:

- 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.
- Dopo 5 giorni, XYZ-3.7 da unstable migra in testing.
- Perciò il 15 giugno entrambe testing e unstable hanno nei loro repository XYZ-3.7.
- 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.
- L'entrata in testing di questa versione nuova in unstable, XYZ-3.8, è pianificata per il 5 luglio.
- Ma il 3 luglio qualcun altro scopre un altro bug critico per il rilascio 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.
- Perciò l'8 luglio testing ha XYZ-3.7, mentre unstable ha XYZ-3.9.
- L'entrata in testing di questa versione nuova, XYZ-3.9, è ora pianificata per il 13 luglio.
- Dato che si usa testing e dato che XYZ-3.7 è affetta da un bug, sarà probabilmente possibile usare XYZ solo dopo il 13 luglio. Cioè, fondamentalmente ci si è ritrovati con una versione guasta di XYZ per circa un mese.

La situazione può complicarsi ulteriormente se, per esempio, XYZ dipende da 4 altri pacchetti. Ciò potrebbe risultare in una distribuzione testing inutilizzabile per mesi. Sebbene lo scenario descritto sopra sia immaginario, cose simili possono verificarsi nella vita reale, anche se raramente.

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

Una delle ragioni principali per cui molte persone scelgono Debian invece di altre distribuzioni Linux è il fatto che richiede molto poca amministrazione. Le persone vogliono un sistema che semplicemente funzioni. In generale si può dire che stable richiede molto poca manutenzione mentre testing e unstable richiedono una manutenzione costante da parte dell'amministratore. Se si usa stable, tutto ciò di cui ci si deve preoccupare è il tenere traccia degli aggiornamenti di sicurezza. Se si usa testing o unstable è una buona idea tenersi al corrente dei nuovi bug scoperti nei pacchetti installati, delle nuove risoluzioni di bug, delle funzionalità introdotte, ecc.

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.

La distribuzione stable è attualmente buster; la prossima versione stable si chiamerà bullseye. Consideriamo il caso particolare di ciò che avverrà quando bullseye verrà rilasciata come nuova versione stable.

- oldstable = stretch; stable = buster; testing = bullseye; 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 bullseye). I pacchetti in stable (cioè buster) 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).
- Quando testing è in freeze, tutte le nuove risoluzioni dei bug introdotte devono essere controllate manualmente dai membri del team per il rilascio. Ciò viene fatto per assicurare che non vi sarà alcun problema serio sconosciuto nella versione testing congelata.
- 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.
- La "testing congelata" senza bug RC viene rilasciata come nuova versione stable. Nel nostro esempio, questo nuovo rilascio stable sarà chiamato bullseye.
- A questo punto si avrà oldstable = buster, stable = bullseye. 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 Al momento sto usando stable. Posso passare a testing o unstable? Se sì, come?

Se si sta attualmente usando `stable`, allora nel file `/etc/apt/sources.list` il terzo campo sarà o «buster» 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».

Attualmente `testing` si chiama `bullseye`. Perciò, se si cambia il terzo campo di `/etc/apt/sources.list` in «bullseye», si userà anche in questo caso `testing`. Però si continuerà ad usare `bullseye` anche quando `bullseye` diventerà `stable`.

`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 `aptitude update` 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 (bullseye). 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/ bullseye 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 `bullseye` 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 "bullseye" allora si userà stable (dato che bullseye sarà allora la nuova distribuzione stable).

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

Se non si è sicuri, la scelta migliore sarebbe la distribuzione stabile.

3.2 E per quanto riguarda Knoppix, Linux Mint Debian Edition, Ubuntu e altre?

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](https://www.debian.org/misc/children-distros) (<https://www.debian.org/misc/children-distros>) disponibile sul sito web di Debian.

3.2.1 So che Knoppix/Linux Mint Debian Edition/Ubuntu/... è basata su Debian. Perciò dopo l'installazione sull'hard disk, posso usare con essa gli strumenti «apt» per i pacchetti?

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.

Knoppix, ad esempio, è una distribuzione Linux progettata per essere avviata da un CD Live mentre Debian è progettata per essere installata sull'hard disk. Knoppix è molto bella se si desidera sapere se un particolare hardware funziona o se si vuole provare a vedere come è un sistema GNU/Linux, ecc.; Knoppix è adatta a scopi dimostrativi mentre Debian è progettata per essere in funzione 24 ore al giorno, 7 giorni su 7. In più sia il numero dei pacchetti disponibili, sia il numero di architetture supportate di Debian sono estremamente di più di quelli di 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 risparmierebbe un sacco di problemi futuri se si installerà subito Debian.

3.2.2 Ho installato Knoppix/Linux Mint Debian Edition/Ubuntu/... sul mio hard disk. Ora ho un problema. Cosa devo fare?

È sconsigliato usare i forum Debian (mailing list o IRC) per chiedere aiuto perché le persone potrebbero dare suggerimenti pensando che si stia usando un sistema Debian e le "soluzioni" fornite potrebbero non essere adatte alla distribuzione che si sta usando; potrebbero persino peggiorare il problema.

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 Sto usando Knoppix/LMDE/Ubuntu/... e ora voglio usare Debian. Come migro a Debian?

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) (<http://en.tldp.org/FAQ/Linux-FAQ/intro.html#DOES-LINUX-RUN-ON-MY-COMPUTER>) per i dettagli.

L'attuale rilascio 10 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*.

Lo sviluppo di distribuzioni binarie di Debian per architetture *hurd-i386* (per kernel GNU Hurd su PC i386 a 32-bit), *mipsel64* (per MIPS a 64 bit in modalità little-endian), *powerpcspe* (port per l'hardware «Signal Processing Engine»), *sparc64* (per processori SPARC a 64 SPARC processors), *sh* (per i processori Hitachi SuperH) e *x32* (per CPU amd64/x86_64 che usano puntatori a 32 bit) è attualmente in corso.

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 stabile 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*.

Il supporto per il port a 32 bit *s390* (*s390*) è stato abbandonato e sostituito in Jessie (Debian 8) da *s390x*. In aggiunta è stato necessario rimuovere i port per IA-64 e Sparc da questo rilascio a causa di un insufficiente supporto da parte degli sviluppatori.

Per ulteriori informazioni sui port disponibili vedere le [pagine dei port sul sito web](https://www.debian.org/ports/) (<https://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 <https://www.debian.org/releases/stable/installmanual>.

4.2 Quali kernel può eseguire Debian GNU/Linux?

Oltre a Linux, Debian fornisce una distribuzione binaria completa per i seguenti kernel per sistemi operativi:

- FreeBSD: fornito attraverso i port *kfreebsd-amd64* e *kfreebsd-i386*, per i PC a 64 bit e 32 bit rispettivamente. Questi port sono stati rilasciati per la prima volta in Debian 6.0 Squeeze come *anteprima tecnologica*. Tuttavia non facevano parte del rilascio Debian 8 Jessie.

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](https://www.debian.org/ports/) (<https://www.debian.org/ports/>).

4.3 Quanto è compatibile Debian con le altre distribuzioni Linux?

Debian developers communicate with other Linux distribution creators in an effort to maintain binary compatibility across Linux distributions. ¹ Most commercial Linux products run as well under Debian as they do on the system upon which they were built.

Debian GNU/Linux aderisce al [Linux Filesystem Hierarchy Standard](http://www.pathname.com/fhs) (<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.

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

¹The [Linux Standard Base](https://wiki.linuxfoundation.org/lsb/start/) (<https://wiki.linuxfoundation.org/lsb/start/>) is a specification for allowing the same binary package to be used on multiple distributions. After Jessie (Debian 8) was released, Debian [abandoned](https://sources.debian.org/src/lsb/9.20170808/debian/README.Debian/) (<https://sources.debian.org/src/lsb/9.20170808/debian/README.Debian/>) the pursuit of LSB compatibility. See this [July 3, 2015 message from Didier Raboud](https://lists.debian.org/4526217.myWFlvm1rM@gyllingar) (<https://lists.debian.org/4526217.myWFlvm1rM@gyllingar>) and the following discussion for background information.

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** (<https://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 Sezione 12.1).

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;
- una completa suite di applicazioni per la rete, inclusi server per protocolli Internet come HTTP (WWW), FTP, NNTP (news), SMTP e POP (posta) e DNS (server dei nomi); database relazionali come PostgreSQL e MySQL; sono inoltre forniti browser web inclusi i vari prodotti Mozilla;
- un insieme completo di applicazioni per l'ufficio, inclusi la suite di produttività LibreOffice, Gnumeric e altri fogli di calcolo, editor WYSIWYG, calendari.

Sono inclusi nella distribuzione più di 57200 pacchetti che spaziano dai server e lettori per le news al supporto per il suono, programmi per FAX, programmi per database e fogli di calcolo, programmi per l'elaborazione delle immagini, comunicazione, rete e utilità per la posta elettronica, server Web e persino programmi per i radioamatori. Altre 930 suite di software sono disponibili come pacchetti Debian, ma non fanno formalmente parte di Debian a causa di restrizioni delle licenze.

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 Sezione 7.4) 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** (<https://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** (<https://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 Sezione 10.2.

5.5 Cosa manca da Debian GNU/Linux?

Esiste un elenco di pacchetti che sono ancora da impacchettare per Debian, l'**elenco Work-Needing and Prospective Packages** (<https://www.debian.org/devel/wnpp/>).

Per maggiori dettagli sull'aggiunta di cose mancanti, si veda Capitolo 13.

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.

L'esecuzione di un'applet Java richiede un browser web con la capacità di riconoscerla ed eseguirla. Diversi browser web disponibili in Debian, come Mozilla o Konqueror, supportano i plug-in Java che abilitano l'esecuzione di applet Java al loro interno.

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

5.8 Come si può verificare di stare usando un sistema Debian e che versione è?

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`. Per una descrizione più prolissa, usare:

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

Per vedere le versioni di tutti i pacchetti installati eseguire:

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

Notare che l'esistenza del programma `dpkg` mostra che si dovrebbe essere in grado di installare pacchetti Debian sul proprio sistema. Tuttavia, dato che sono stati fatti port del programma su molti altri sistemi operativi e architetture, questo non è più un metodo affidabile per determinare se un sistema è 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.

- Quasi tutto il software in Debian gestisce UTF-8 come insieme di caratteri. Insieme di caratteri datati, come ISO-8859-1 o ISO-8859-2, dovrebbero essere considerati obsoleti.
- Attualmente è disponibile il supporto per le pagine di manuale nelle lingue tedesco, spagnolo, francese, ungherese, italiano, giapponese, coreano, olandese, polacco, portoghese, russo, turco e cinese attraverso i pacchetti `manpages-LANG` (dove `LANG` è il codice ISO a due lettere della nazione). Per accedere alla pagina di un manuale NLS, l'utente deve impostare la variabile `LC_MESSAGES` della shell alla stringa appropriata.

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** (<https://cr.yp.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** (<https://cr.yp.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 è disponibile per GNU/Linux dal sito web di Google ma, non solo non è software libero, è assolutamente non distribuibile da terze parti. `googleearth-package` (nella sezione `contrib`) può tuttavia essere d'aiuto nell'uso di questo 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** (<https://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 Sezione 6.5.1). Oltre a queste c'è la distribuzione "oldstable" (che è semplicemente quella che ha preceduto "stable") e quella "experimental".

Experimental è usata per i pacchetti che sono ancora in fase di sviluppo e con un alto rischio di danneggiamento del sistema. Viene usata da sviluppatori che vogliono studiare e testare software recentissimo. Gli utenti non dovrebbero usare pacchetti provenienti da lì perché questi possono essere pericolosi e creare danni anche per le persone più esperte.

Per un aiuto al momento di scegliere una distribuzione Debian, si veda Capitolo 3.

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 `buster` (ovvero Debian GNU/Linux 10) e `testing` è un collegamento simbolico a `bullseye`. Questo significa che `buster` è la distribuzione `stable` attuale e che `bullseye` è la distribuzione `testing` attuale.

`unstable` è un collegamento simbolico permanente a `sid`, dato che `sid` è sempre la distribuzione `unstable` (si veda Sezione 6.3).

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

Oltre a `buster` e `bullseye`, altri nomi in codice che sono già stati usati sono: `buzz` per il rilascio 1.1, `rex` per il rilascio 1.2, `bo` per il rilascio 1.3.x, `hamm` per il rilascio 2.0, `slink` per il rilascio 2.1, `potato` per il rilascio 2.2, `woody` per il rilascio 3.0, `sarge` per il rilascio 3.1, `etch` per il rilascio 4.0, `lenny` per il rilascio 5.0 e `squeeze` per il rilascio 6.0, `wheezy` per il rilascio 7, `jessie` per il rilascio 8 e `stretch` per il rilascio 9.

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* (Debian 1.1) was the spaceman Buzz Lightyear,
- *rex* (Debian 1.2) was the tyrannosaurus,
- *bo* (Debian 1.3) was Bo Peep, the girl who took care of the sheep,
- *hamm* (Debian 2.0) was the piggy bank,

- *slink* (Debian 2.1) was Slinky Dog, the toy dog,
- *potato* (Debian 2.2) was, of course, Mr. Potato,
- *woody* (Debian 3.0) was the cowboy,
- *sarge* (Debian 3.1) was the sergeant of the Green Plastic Army Men,
- *etch* (Debian 4.0) was the toy whiteboard (Etch-a-Sketch),
- *lenny* (Debian 5.0) was the toy binoculars,
- *squeeze* (Debian 6) was the name of the three-eyed aliens,
- *wheezy* (Debian 7) era il pinguino di gomma con un farfallino rosso,
- *jessie* (Debian 8) era la cowgirl che faceva lo yodel,
- *stretch* (Debian 9) era la piovra giocattolo di gomma con ventose sulle sue otto lunghe braccia,
- *buster* (Debian 10) era il cagnolino di Andy,
- *bullseye* (Debian 11) era il cavallo di legno di Woody,
- *bookworm* (Debian 12) era un verme giocattolo verde con una torcia incorporata che amava leggere libri,
- *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** (https://www.debian.org/social_contract#guidelines) e sono tutti liberamente utilizzabili e distribuibili.

¹When the present-day *sid* did not exist, the FTP site organization had one major flaw: there was an assumption that when an architecture is created in the current *unstable*, it will be released when that distribution becomes the new *stable*. For many architectures that isn't the case, with the result that those directories had to be moved at release time. This was impractical because the move would chew up lots of bandwidth.

The archive administrators worked around this problem for several years by placing binaries for unreleased architectures in a special directory called "sid". For those architectures not yet released, the first time they were released there was a link from the current *stable* to *sid*, and from then on they were created inside the *unstable* tree as normal. This layout was somewhat confusing to users.

With the advent of package pools (see Sezione 6.10), binary packages began to be stored in a canonical location in the pool, regardless of the distribution, so releasing a distribution no longer causes large bandwidth consumption on the mirrors (there is, however, a lot of gradual bandwidth consumption throughout the development process).

- **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.

6.5 Cosa contiene la distribuzione testing?

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

Devono essere sincronizzati in tutte le architetture per le quali sono stati compilati e non devono avere dipendenze tali da renderli non installabili; devono inoltre avere meno bug critici per il rilascio delle versioni attualmente in unstable. In questo modo, si auspica che «testing» sia sempre vicina ad essere una candidata al rilascio.

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

6.5.1 Cosa dire di "testing"? Come viene "congelata"?

Quando la distribuzione «testing» è abbastanza matura, il responsabile del rilascio inizia a «congelarla». I normali ritardi di diffusione vengono aumentati per assicurare che entri da «unstable» in «testing» il minor numero di bug possibile.

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** (<https://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** (<https://bugs.debian.org/release-critical/>). Un comune obiettivo di rilascio è **NoRCBugs** (<https://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** (https://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** (<https://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 Sezione 6.10.

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`?

Within each of the major directory trees², there are three sets of subdirectories containing index files.

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](https://www.debian.org/releases) (<https://www.debian.org/releases>). Per il rilascio attuale, si veda Sezione 4.1.

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 Sezione 6.10) 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-mio-pacchetto`.

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`).

²`dists/stable/main`, `dists/stable/contrib`, `dists/stable/non-free`, and `dists/unstable/main/`, etc.

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 mio pacchetto` 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

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 <https://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) (<https://wiki.debian.org/HowToSetupADebianRepository>).

³Historically, packages were kept in the subdirectory of `dists` corresponding to which distribution contained them. This turned out to cause various problems, such as large bandwidth consumption on mirrors when major changes were made. This was fixed with the introduction of the package pool.

The `dists` directories are still used for the index files used by programs like apt.

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 Capitolo 8 e Capitolo 9.

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:

- *Pacchetti binari*, che contengono eseguibili, file di configurazione, pagine man/info, informazioni sul copyright ed altra documentazione. Questi pacchetti sono distribuiti in un formato archivio specifico di Debian (si veda Sezione 7.2); sono solitamente caratterizzati dall'estensione «.deb». I pacchetti binari possono essere spaccettati usando l'utilità Debian `dpkg` (opzionalmente tramite un frontend come **aptitude**); i dettagli sono forniti nella sua pagina di manuale.
- *Pacchetti sorgente*, che consistono in un file `.dsc` che descrive il pacchetto sorgente (inclusi i nomi dei file seguenti), un file `.orig.tar.gz` che contiene il sorgente originale non modificato nel formato tar compresso con gzip e solitamente un file `.diff.gz` che contiene le modifiche specifiche di Debian fatte al sorgente originale. L'utilità `dpkg-source` pacchetta e spacchetta gli archivi sorgente Debian; i dettagli sono forniti nella sua pagina di manuale. (Il programma **apt-get** può essere usato come frontend per `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 Sezione 7.9 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`

Si noti che `pippo` dovrebbe essere il nome del pacchetto. Si può controllare il nome del pacchetto associato ad un particolare file di archivio Debian (`file.deb`) in uno dei modi seguenti:

- 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.

La parte `AAA` identifica il processore per cui è stato compilato il pacchetto. È solitamente `amd64`, che si riferisce ai chip AMD64, Intel 64 o VIA Nano. Per altre possibilità si veda la struttura delle directory dei siti FTP di Debian in Sezione 6.7. Per dettagli, si veda la descrizione delle «architetture Debian» nella pagina di manuale `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 Sezione 12.1.

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: https://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 Sezione 7.3.

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 Sezione 7.7.

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 Sezione 12.1).

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

`postinst` Questo script tipicamente completa ogni configurazione richiesta dal pacchetto `pippo` una volta che `pippo` è stato spaccettato dal suo archivio Debian («`.deb`»). Spesso gli script «`postinst`» richiedono all'utente degli input o lo avvertono che se accetta le impostazioni predefinite deve ricordarsi di tornare indietro e riconfigurare il pacchetto se la situazione lo richiede. Molti script «`postinst`» eseguono poi tutti i comandi necessari ad avviare o riavviare un servizio una volta che un nuovo pacchetto è stato installato o aggiornato.

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 Sezione 7.8.)

Attualmente tutti i file di controllo si possono trovare nella directory `/var/lib/dpkg/info`. I file attinenti al pacchetto `pippo` iniziano con il nome «pippo» ed hanno le estensioni «preinst», «postinst», ecc., a seconda della funzione. Il file `pippo.list` nella stessa directory elenca tutti i file installati con il pacchetto `pippo`. (Si noti che la posizione di questi file è un dato tecnico interno di `dpkg`; non ci si dovrebbe fare affidamento.)

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*.
Comprendono altri pacchetti senza i quali il sistema non funziona bene o non sarebbe utilizzabile. Tra questi *NON* sono inclusi Emacs o X o TeX o qualsiasi altra grossa applicazione. Questi pacchetti costituiscono solo l'infrastruttura di base.
- **Standard:** pacchetti comuni su qualsiasi sistema Linux, compreso un ragionevolmente piccolo ma nemmeno troppo limitato sistema a caratteri. Sono inclusi strumenti per inviare posta elettronica (con `mutt`) e scaricare file da server FTP.
Questo è ciò che viene installato in modo predefinito se l'utente non seleziona nient'altro. Non include molte grosse applicazioni, ma include l'interprete Python e qualche software server come OpenSSH (per amministrazione da remoto) ed Exim (per la consegna della posta, anche se può essere configurato per la sola consegna locale). Include anche della documentazione generica comune che può essere utile per la maggior parte degli utenti.
- **Optional (Opzionale):** pacchetti che comprendono tutto quello che si può voler installare senza nemmeno sapere cos'è, o se non si hanno delle necessità particolari.
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?

Un pacchetto virtuale è un nome generico che si applica ad ognuno dei componenti di un gruppo di pacchetti, che forniscono funzionalità di base simili. Per esempio, entrambi i programmi `konqueror` e `firefox-esr` sono browser web e dovrebbero quindi soddisfare qualsiasi dipendenza di un programma

che richiede un browser web sul sistema, per poter funzionare o essere utile. Si dice quindi che entrambi forniscono il «pacchetto virtuale» chiamato `www-browser`.

Similmente, `exim4` e `sendmail` forniscono entrambi la funzionalità di agente di trasporto posta. Si dice quindi che forniscono il pacchetto virtuale «mail-transport-agent». Se uno qualsiasi dei due è installato, allora ogni pacchetto che dipende dall'installazione di un `mail-transport-agent` vedrà le proprie dipendenze soddisfatte dalla presenza di questo pacchetto virtuale.

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 Sezione 11.11.

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.

Informazioni più dettagliate sull'utilizzo di ognuno di questi termini possono essere trovate nel manuale Debian Policy, sezione 7.2 "Binary Dependencies" (vedere Sezione 12.1).

7.10 Cosa significa Pre-Depends?

«Pre-Depends» è una dipendenza speciale. Per la maggior parte dei pacchetti, `dpkg` spacchetta il loro file di archivio (ovvero il loro file `.deb`) indipendentemente dal fatto che i file da cui dipendono siano o meno sul sistema. Semplificando, spacchettare vuol dire che `dpkg` estrae i file da installare dall'archivio e li mette al loro posto nel file system. Se quei pacchetti *dipendono* dall'esistenza di qualche altro pacchetto sul sistema, `dpkg` si rifiuta di completare l'installazione (eseguendo l'azione «configura»), finché gli altri pacchetti non sono installati.

Tuttavia, per alcuni pacchetti, `dpkg` si rifiuta persino di spacchettarli 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;
- **rimuovi**: l'utente vuole che sia rimosso il pacchetto, ma non i file di configurazione esistenti;
- **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 \* > 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 vogliano compilare i pacchetti binari che producono.

I pacchetti sorgente sono distribuiti sulla maggior parte dei mirror dove si possono ottenere i pacchetti binari. Se si imposta il proprio `sources.list(5)` di APT in modo che includa le righe «deb-src» appropriate, sarà possibile scaricare facilmente qualsiasi pacchetto sorgente eseguendo

```
apt-get source pippo
```

Per aiutare nella reale compilazione del pacchetto sorgente, i pacchetti sorgente Debian forniscono il cosiddetto meccanismo di dipendenze di compilazione. Significa che il manutentore del pacchetto sorgente mantiene una lista di altri pacchetti che sono richiesti per compilare il suo pacchetto. Per vedere come questo sia utile, si esegua

```
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 (Sezione 7.13), 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`.

Se si vuole solo compilare il pacchetto, si può entrare nella directory `pippo-versione` e lanciare il comando

```
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?

Per maggiori dettagli al riguardo si legga la New Maintainers' Guide, (Guida per il nuovo Maintainer) disponibile nel pacchetto `maint-guide` o su <https://www.debian.org/doc/devel-manuals#maint-guide>, oppure la «Guide for Debian Maintainers», disponibile nel pacchetto `debmake-doc` o su <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.

Si veda il **Capitolo 2. Gestione dei pacchetti Debian** (<https://www.debian.org/doc/manuals/debian-reference/ch02.en.html>) della **Debian reference** (<https://www.debian.org/doc/manuals/debian-reference/>) per maggiori informazioni sulle utilità di gestione dei pacchetti di Debian. Tale documento è disponibile in diverse lingue e formati; si veda la **voce della Debian Reference nella panoramica dei manuali utenti del DDP** (<https://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 Sezione 7.6) 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 Sezione 7.6) 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 "ciccìa" (o un gruppo di file chiamati "ciccìa*") da un archivio Debian: `dpkg --fsys-tarfile pippo_VVV-RRR.deb | tar -xf - 'ciccìa*'.`
- 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 --get-frontend 'pippo*'`.

8.1.2 APT

APT è l'*Advanced Package Tool* (strumento avanzato per i pacchetti), un'interfaccia avanzata per il sistema di gestione dei pacchetti di Debian che fornisce il programma **apt-get**. Fornisce strumenti a riga di comando per cercare e gestire pacchetti e per interrogare informazioni su di essi, oltre all'accesso a basso livello a tutte le funzionalità della libreria `libapt-pkg`. Per maggiori informazioni vedere la Guida utente in `/usr/share/doc/apt-doc/guide.html/index.html` (è necessario installare il pacchetto `apt-doc`).

A partire da Debian Jessie per alcuni comandi **apt-get** e **apt-cache** di uso frequente esiste un equivalente tramite il nuovo binario **apt**. Ciò significa che comandi popolari come **apt-get update**, **apt-get install**, **apt-get remove**, **apt-cache search** o **apt-cache show** ora possono anche essere chiamati semplicemente con **apt**, ad esempio con **apt update**, **apt install**, **apt remove**, **apt search** o **apt show**. Di seguito viene riportata una panoramica dei comandi vecchi e nuovi:

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 provides a simple way to retrieve and install packages from multiple sources using the command line. Unlike **dpkg**, **apt-get** does not understand .deb files, it works with the packages proper name and can only install .deb archives from a source specified in `/etc/apt/sources.list`. **apt-get** will call **dpkg** directly after downloading the .deb archives¹ from the configured sources.

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

- Per aggiornare l'elenco dei pacchetti conosciuti dal sistema, si può eseguire:

```
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 pipbo dal sistema, si esegua:

```
apt remove pippo
```

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

```
apt purge pipbo
```

- Per elencare tutti i pacchetti per i quali sono disponibili versioni più nuove, eseguire:

```
apt list --upgradable
```

¹Notice that there are ports that make this tool available with other package management systems, like Red Hat package manager, also known as **rpm**

- 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
```

(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.)

Si noti che si deve aver fatto il login come `root` per eseguire qualsiasi comando che modifichi i pacchetti.

Si noti che **apt-get** ora installa in modo predefinito anche i pacchetti raccomandati ed è, per la sua robustezza, il programma consigliato per la gestione dei pacchetti dalla console, per eseguire installazioni di sistema e per aggiornamenti principali del sistema.

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
```

- Per stampare informazioni dettagliate sulle versioni disponibili di un pacchetto e le sue dipendenze inverse:

```
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 Sezione [8.1.5](#).
- **aptitude** in modalità a schermo intero ha una funzionalità **su** incorporata e può essere eseguito da un utente normale. Richiama **su** (e chiederà la password di root, se esiste) quando sono veramente necessari i privilegi di amministrazione.

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`.

Si noti che **aptitude** è il programma consigliato per la gestione quotidiana dei pacchetti dalla console.

Per maggiori informazioni, si legga la pagina di manuale `aptitude(8)` e si installi il pacchetto `aptitude-doc`.

8.1.4 synaptic

synaptic è un gestore di pacchetti grafico. Permette di installare, aggiornare e rimuovere pacchetti software in modo amichevole. Oltre alla maggior parte delle funzionalità offerte da **aptitude**, ha anche una funzionalità per modificare l'elenco dei repository usati e supporta la navigazione di tutta la documentazione disponibile relativa ad un pacchetto. Si veda il [sito web di synaptic](https://www.nongnu.org/synaptic/) (<https://www.nongnu.org/synaptic/>) per maggiori informazioni.

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**.

L'installatore Debian tipicamente installa automaticamente le attività associate con un sistema standard ed un ambiente desktop. L'ambiente desktop specifico installato dipende dal supporto CD/DVD usato, solitamente è il desktop GNOME (attività `gnome-desktop`). Inoltre, a seconda delle selezioni dell'utente durante il processo di installazione, possono essere installate automaticamente delle attività. Per esempio, se si è selezionata una lingua diversa dall'inglese, anche le attività associate ad essa vengono automaticamente installate, e se l'installatore riconosce che l'installazione viene fatta su un sistema portatile viene installata anche l'attività `laptop`.

8.1.6 Altri strumenti per la gestione dei pacchetti

8.1.6.1 dpkg-deb

Questo programma manipola i file archivio Debian (`.deb`). Quelli che seguono sono alcuni degli usi comuni.

- 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.

Per sapere lo stato dei pacchetti i cui nomi corrispondono a qualsiasi modello che inizia con «pippo» eseguire il comando:

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

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

```
dpkg --status nomepacchetto
```

8.4 Come mostrare i file di un pacchetto installato?

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

```
dpkg --getfiles 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 --get-selections 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/buster/`). 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 --get-selections` di trovare i file nei pacchetti che non sono attualmente installati sul proprio sistema.

- `apt-file search pippo`

Se si installa il pacchetto `apt-file`, simile al precedente, esso cerca i file che contengono nel loro nome di percorso completo la sottostringa o l'espressione regolare `pippo`. Il vantaggio rispetto all'esempio precedente è che non c'è bisogno di recuperare i file `Contents-ARCH.gz` dato che ciò viene fatto automaticamente per tutti le fonti definite in `/etc/apt/sources.list` quando viene eseguito (come root) `apt-file update`.

8.6 Perché «pippo-data» non viene rimosso quando viene disinstallato «pippo»? Come ci si può assicurare che vengano eliminati i vecchi pacchetti di librerie non utilizzati?

Alcuni pacchetti sono suddivisi nel programma («pippo») e nei dati («pippo-data») (o in «pippo» e «pippo-doc»). Questo sistema vale in Debian per molti giochi, applicazioni multimediali e dizionari ed è stato introdotto dato che alcuni utenti possono voler accedere ai dati grezzi senza installare il programma o perché il programma può essere eseguito senza i dati rendendo così «pippo-data» opzionale.

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 questi casi, «pippo-data» non dipende da «pippo», perciò la maggior parte degli strumenti di gestione dei pacchetti non lo rimuove automaticamente quando viene rimosso il pacchetto «pippo». La stessa cosa si verifica per i pacchetti di libreria. Ciò è necessario per evitare dipendenze cicliche. Se però si usa **apt-get** (vedere Sezione 8.1.2) o **aptitude** (vedere Sezione 8.1.3) come strumento di gestione dei pacchetti, essi tengono traccia dei pacchetti installati automaticamente e danno la possibilità di rimuoverli quando sul sistema non rimane alcun pacchetto che ne abbia bisogno.

Capitolo 9

Mantenere il sistema Debian aggiornato

Uno degli scopi di Debian è quello di fornire un coerente percorso di aggiornamento ed un sicuro processo di aggiornamento. Viene sempre fatto tutto il possibile per rendere semplice l'aggiornamento a nuovi rilasci. Nel caso ci siano note importanti da aggiungere al processo di aggiornamento, i pacchetti avvisano l'utente e spesso forniscono una soluzione ad un possibile problema.

Si dovrebbero inoltre leggere le Note di rilascio (Release Notes), documento che descrive i dettagli degli specifici aggiornamenti. È disponibile sul sito web Debian all'indirizzo <https://www.debian.org/releases/stable/releasenotes> e viene anche fornito sui CD, DVD e dischi Blu-Ray Debian.

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 Sezione 8.1.3.

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 <https://www.debian.org/mirror/list>.

Oppure si può usare il servizio di ridirezionamento `httpredir.debian.org` che mira a risolvere il problema della scelta di un mirror Debian. Usa la posizione geografica dell'utente e altre informazioni per scegliere il mirror migliore che può servire i file. Per sfruttarlo usare una fonte simile alla seguente:

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

Ulteriori dettagli a questo proposito possono essere trovati nella pagina di manuale `sources.list(5)`. Per aggiornare il proprio sistema dalla riga di comando, eseguire

```
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** (<https://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 Sezione 8.1.2).

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 Sezione 9.1.1.

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 Sezione 8.1.2.

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.

Il programma **mirror** è generalmente utile per aggiornare copie locali di alberi di directory remoti. I file prelevati non devono necessariamente essere file Debian. (Dato che **mirror** è uno script in Perl, può anche essere eseguito su sistemi non Unix.) Sebbene il programma **mirror** fornisca meccanismi per escludere file con nomi che corrispondono a stringhe definite dall'utente, questo programma è più utile quando l'obiettivo è scaricare interi alberi di directory, piuttosto che pacchetti specifici.

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 Come si può tenere un registro dei pacchetti che sono stati aggiunti al sistema? Vorrei sapere quali aggiornamenti e rimozioni di pacchetti sono stati fatti!

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
```

). Se si desidera tenere un registro di tutte le proprie invocazioni di **dpkg** (anche quelle fatte attraverso frontend come **aptitude**), si può aggiungere

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

al proprio `/etc/dpkg/dpkg.cfg`. Assicurarsi che il file di registro creato venga periodicamente ruotato. Se si usa **logrotate**, questo può essere fatto creando un file `/etc/logrotate.d/dpkg` contenente le seguenti righe

```
/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.

C'è solo un problema comune: le librerie C di Debian sono compilate con il più recente rilascio *stabile* degli header del **kernel**. Se si ha bisogno di compilare un programma con gli header del kernel più nuovi rispetto a quelli nel ramo stabile, allora si dovrebbe aggiornare il pacchetto contenente gli header (`linux-libc-dev`) oppure usare i nuovi header da un albero di directory non pacchettizzato del kernel più recente. Cioè, se i sorgenti del kernel sono in `/usr/src/linux`, si dovrebbe aggiungere `-I/usr/src/linux/include/` alla propria riga di comando quando si compila.

10.2 Quali strumenti fornisce Debian per costruire kernel personalizzati?

Per gli utenti che desiderano (o devono) compilare un kernel personalizzato è raccomandato l'uso del target per pacchetti Debian incluso con le più recenti versioni del sistema di compilazione del kernel. Dopo aver configurato il kernel, basta eseguire il seguente comando:

```
make deb-pkg
```

Il nuovo pacchetto del kernel verrà creato nella directory un livello più in alto dell'albero dei sorgenti del kernel e può essere installato usando `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?

Un file di configurazione contenente i moduli da caricare manualmente all'avvio viene mantenuto in `/etc/modules`. Tuttavia raramente è necessario modificarlo.

Altra configurazione per i moduli viene mantenuta nella directory `/etc/modprobe.d/`. Ulteriori informazioni sul formato di questi file possono essere trovate nella pagina di manuale `modprobe.conf(5)`.

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

Sì. Lo script `linux-image-xxx.prerm` 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 Come si possono ottenere ulteriori informazioni sui pacchetti Linux per Debian?

Ulteriori informazioni sono contenute nel [Debian Linux Kernel Handbook](https://kernel-team.pages.debian.net/kernel-handbook/) (<https://kernel-team.pages.debian.net/kernel-handbook/>).

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 Come si avvia un sistema Debian?

Like all Unices, Debian boots up by executing the program `init`. Like most Linux distributions, a default Debian system uses `systemd` as the implementation of `init`. Traditional System-V style `init` and other methods are also supported. ¹

Per controllare l'ordine in cui i servizi sono avviati i sistemi Unix tradizionali in stile System V usano i *runlevel*. Questi in `systemd` sono rimpiazzati da *target*. Per visualizzare il target predefinito al quale `systemd` porta il sistema, eseguire il comando

```
systemctl get-default
```

Durante l'avvio `systemd` fa partire i servizi o gli altri target elencati in file del target predefinito `/lib/systemd/system/default.target`. I file per questi servizi e target vengono installati e il servizio è *abilitato* durante l'installazione dei pacchetti Debian. Se si desidera specificatamente di non avviare un servizio all'avvio di sistema, invece di rimuovere il pacchetto corrispondente si può eseguire il comando

```
systemctl disable servizio.service
```

usando il nome del file di servizio installato in `/lib/systemd/system` (solitamente basato sul nome del pacchetto).

Il file di servizio `/lib/systemd/rc.local.service` fornisce un modo semplice per eseguire dopo l'avvio script personalizzati nel file `/etc/rc.local`, in modo simile a ciò che viene offerto nei sistemi Debian che eseguono un `init` in stile System V. Attenzione: questo script fallisce se cerca di interagire con la console, ad esempio per chiedere una password utente o per cercare di pulire lo schermo.

Si può controllare lo stato di ogni servizio con il comando

```
service pacchetto status
```

. Per avviare o fermare un servizio eseguire

```
service pacchetto start
```

e

```
service pacchetto stop
```

. Il comando `service` funziona con qualsiasi sistema `init` gestito su un sistema Debian, non solo con `systemd`. Se tuttavia si preferisce usare lo stesso comando su qualsiasi sistema Linux con `systemd`, per controllare lo stato eseguire

```
systemctl status pacchetto.service
```

per ottenere le stesse informazioni.

Per ulteriori informazioni su `systemd` per Debian vedere <https://wiki.debian.org/systemd>.

11.6 Cosa ne è dell'init System V tradizionale in Debian?

Debian gestisce l'avvio attraverso l'`init` tradizionale System V attraverso il pacchetto `sysvinit-core`. Il file di configurazione per l'`init` System V (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.

¹In 2014, Debian changed its default `init` system from System V `init` to `systemd`. Debian 8 "jessie" in April 2015 was the first release to ship with `systemd` as default `init`. Four decisions (<https://www.debian.org/devel/tech-ctte#status>) of the Debian Technical Committee were involved: Bug #727708 (<https://lists.debian.org/20140211193904.GX24404@rzlab.ucr.edu>) 2014-02-11: "The committee decided that the default `init` system for Linux architectures in jessie should be `systemd`." Bug #746715 (<https://lists.debian.org/20140801023630.GF12356@teltox.donarmstrong.com>) 2014-08-01: "The technical committee expects maintainers to continue to support the multiple available `init` systems", and merge reasonable contributions. Bug #746578 (<https://lists.debian.org/20141116001628.G032192@teltox.donarmstrong.com>) 2014-11-15: "The committee decided that `systemd-shim` should be the first listed alternative dependency of `libpam-systemd` instead of `systemd-sysv`." This decision made it easier to keep running a non-`systemd` Debian system. Bug #762194 (<https://lists.debian.org/201502.61064.527547.410074@chiark.greenend.org.uk>) 2017-11-04: "On automatic `init` system switching on upgrade"

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**.

11.7 Esistono ancora altri modi di avviare un sistema Debian?

Se ci si trova bene con l'init System V, ma non piacciono i collegamenti `/etc/rc?.d/*` si può installare il pacchetto `file-rc`. Questo converte i collegamenti in un singolo file di configurazione `/etc/runlevel.conf`

Se non piace né System V né `systemd`, si può provare `openrc`, `runit` o `daemontools`.

11.8 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.

11.9 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)`.

11.10 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 [PREFIXOPERCORSO] > 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.
- `PREFIXOPERCORSO` 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)`.

11.11 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 <https://www.debian.org/releases/stable/installmanual>.

- The Debian GNU/Linux reference covers many aspects of system administration through shell-command examples. Basic tutorials, tips, and other information are provided for many different topics ranging from system administration to programming.

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

- Documentation developed by the Debian Documentation Project. It is available at <https://www.debian.org/doc> and includes user guides, administration guides and security guides for the Debian GNU/Linux operating system.

- 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/) (<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 Sezione 5.9.)

- 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.

Il principale vantaggio rispetto alle originali pagine di manuale è che viene usato un sistema ad ipertesto. Comunque, *non* richiede il WWW; `info` può essere eseguito da una semplice console testuale. È stato progettato da Richard Stallman ed ha preceduto il 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](https://www.debian.org/support) (<https://www.debian.org/support>).

12.2.1 Mailing-list

Ci sono molte [mailing-list relative a Debian](https://www.debian.org/MailingLists/) (<https://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](https://www.debian.org/MailingLists/subscribe) (<https://www.debian.org/MailingLists/subscribe>). Ci si può anche disiscrivere utilizzando un [modulo WWW](https://www.debian.org/MailingLists/unsubscribe) (<https://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](https://www.debian.org/MailingLists/#disclaimer) (<https://www.debian.org/MailingLists/#disclaimer>) per maggiori informazioni.

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

12.2.1.1 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](https://www.debian.org/MailingLists/#ads) (<https://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.

- Make sure that you are using the proper list. *Never* post your (un)subscription requests to the mailing list itself.¹
- Vedere la sezione Sezione 12.5 per indicazioni sulla segnalazione di bug.

12.2.2 Forum web

debianHELP (<https://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** (<https://wiki.debian.org/>) in costante evoluzione.

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** (<https://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** (<https://search.debian.org/>).
- **Google Groups** (<https://groups.google.com/>): un motore di ricerca per gruppi di discussione.

Per esempio, per cercare che esperienza ha avuto la gente nel trovare dei driver per le schede grafiche NVIDIA in Debian, si provi a cercare l'espressione `NVIDIA Linux driver`. Questo mostrerà tutti i messaggi che contengono queste stringhe, ovvero quelli dove la gente ha discusso questi argomenti. Se si aggiunge `Debian` a quella stringa di ricerca, si avranno inoltre i messaggi attinenti specificatamente a Debian.

- Tutti i comuni motori di ricerca del web, come **DuckDuckGo** (<https://duckduckgo.com/>) o **Google** (<https://www.google.com/>), purché si usino i termini di ricerca corretti.

Per esempio, cercare la stringa «`evince`» fornisce una spiegazione più dettagliata di questo pacchetto rispetto alla breve descrizione nel suo file di controllo.

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** (https://www.debian.org/social_contract) "Noi non nasconderemo i problemi".

¹Use the `debian-list-subject-REQUEST@lists.debian.org` address for that.

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 <https://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 <https://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 aspetti 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 (<https://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?** (<https://www.debian.org/intro/help>).

13.1 Come si può diventare un membro o uno sviluppatore Debian?

Lo sviluppo di Debian è aperto a tutti e i nuovi utenti con le giuste capacità o con la volontà di imparare sono necessari per mantenere pacchetti già esistenti resi «orfani» dai loro precedenti manutentori, per sviluppare nuovi pacchetti, per scrivere documentazione, per tradurre, per aiutare con il sito web Debian, per fornire supporto agli utenti, ecc.

La descrizione su come diventare un membro di Debian può essere trovata all'**Angolo del nuovo membro** (<https://www.debian.org/devel/join/newmaint>) sul sito web Debian.

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** (<https://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** (<https://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: <https://www.spi-inc.org/>.

13.3.2 Altre organizzazioni

Ci sono diverse organizzazioni create in diverse nazioni che curano beni per conto di Debian. La [pagina sulle donazioni](https://www.debian.org/donations) (<https://www.debian.org/donations>) elenca le organizzazioni fidate a cui le singole persone possono fare donazioni. Al momento della stesura di questo documento ce ne sono tre: [Verein zur Förderung Freier Informationen & Software](http://www.ffis.de/) (<http://www.ffis.de/>) (in Germania), la [Debian France Association](https://france.debian.net/) (<https://france.debian.net/>) (in Francia) e debian.ch (<https://debian.ch/>) (Svizzera e Principato del Liechtenstein). Ulteriori organizzazioni affiliate in altre nazioni sono elencate nella pagina [Organizzazioni](https://wiki.debian.org/Teams/Auditor/Organizations) (<https://wiki.debian.org/Teams/Auditor/Organizations>) del 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?

Sì. Distribuzioni derivate da Debian vengono attualmente create sia in stretta collaborazione con il progetto Debian stesso, sia da terze parti. Si può usare l'infrastruttura [Debian Pure Blends](https://www.debian.org/blends/) (<https://www.debian.org/blends/>) per lavorare insieme a Debian; [DebianEdu/Skoletlinux](https://wiki.debian.org/DebianEdu/) (<https://wiki.debian.org/DebianEdu/>) è un progetto di questo tipo.

Ci sono diverse altre distribuzioni derivate da Debian già sul mercato, come grml, LMDE (Linux Mint Debian Edition, Knoppix e Ubuntu, che sono indirizzate ad un differente tipo di pubblico rispetto all'originale Debian GNU/Linux, ma usano la maggior parte dei nostri componenti nei loro prodotti.

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 Sezione [11.9](#).

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.

Tutto ciò viene fatto costantemente. Per la prima voce viene usato un insieme di opzioni di compilazione per il rafforzamento della sicurezza che cerca di prevenire attacchi noti, come «stack smashing», posizioni prevedibili di valori in memoria, ecc. L'obiettivo è di coprire almeno tutti i pacchetti che fanno parte dell'installazione base, oltre ai pacchetti che è stato necessario aggiornare attraverso un Avvertimento di sicurezza a partire dal 2006. Al momento della stesura di questo documento, circa 400 pacchetti sono stati modificati da quando è stato iniziato questo sforzo. Tutti i problemi sono **tracciati nel 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 Sezione 5.9.

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

Un sistema Debian completo su altre architetture. Notare che benché alcune architetture vengano dismesse per un dato rilascio, ci potrebbe comunque essere un modo per installarle e aggiornarle usando la più recente `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** (<https://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](https://perens.com/) (<https://perens.com/>);
- le Linux FAQ di [Ian Jackson](https://www.chiark.greenend.org.uk/~ijackson/) (<https://www.chiark.greenend.org.uk/~ijackson/>);
- gli [archivi delle mailing-list Debian](https://lists.debian.org/) (<https://lists.debian.org/>);
- il manuale per programmatori di dpkg e il manuale Debian Policy (si veda Sezione [12.1](#));
- 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) (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 Capitolo [3](#)).

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 o inviare una segnalazione di bug di tipo wishlist sul pacchetto [debian-faq](https://bugs.debian.org/debian-faq) (<https://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 <https://www.debian.org/doc/FAQ/>.

It is also available for download in plain text, HTML, and PDF formats at <https://www.debian.org/doc/user-manuals#faq>. Also, there are several translations there.

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

The original XML files used to create this document are also available in `debian-faq`'s source package, or in GIT at: git@salsa.debian.org:ddp-team/debian-faq.git and <https://salsa.debian.org/ddp-team/debian-faq>.

16.4 Formato del documento

This document was written using the DocBook XML DTD. This system enables us to create files in a variety of formats from one source, e.g. this document can be viewed as HTML, plain text, TeX DVI, PostScript, PDF, or GNU info.

Indice analitico

P

packages

- apt, [39](#)
- apt-doc, [38](#), [39](#)
- apt-file, [42](#)
- apt-move, [44](#)
- aptitude-doc, [40](#)
- binutils, [29](#)
- build-essential, [20](#)
- daemontools, [22](#), [51](#)
- dbndns, [22](#)
- debian-faq, [65](#)
- debian-faq-de, [65](#)
- debian-faq-fr, [65](#)
- debian-policy, [55](#)
- debian-reference, [55](#)
- debmake-doc, [36](#)
- djbdns, [22](#)
- doc-debian, [56](#), [58](#)
- doc-linux, [55](#)
- dpkg-dev, [35](#)
- ekiga, [22](#)
- ezmlm-idx, [22](#)
- fakeroot, [35](#)
- file-rc, [51](#)
- gcc, [29](#)
- gnash, [22](#)
- googleearth-package, [22](#)
- kbd, [49](#)
- kernel-package, [20](#)
- labplot, [9](#)
- libc6, [34](#)
- libpaper1, [49](#)
- linux-libc-dev, [47](#)
- linux-source-versione, [47](#)
- login, [52](#)
- lsb-release, [11](#)
- maint-guide, [36](#)
- mlmmj, [22](#)
- ndiswrapper, [22](#)
- news-reader, [53](#)
- openrc, [51](#)
- publicfile-installer, [22](#)
- qmail, [22](#)
- reportbug, [58](#)
- runit, [51](#)
- swfdec, [22](#)
- tin, [53](#)
- trn, [53](#)
- ucspi-tcp, [22](#)