

L'indispensabile sulla gestione dei pacchetti Debian

Daniele Venzano

20 dicembre 2003

Indice

1	Introduzione	2
1.1	Informazioni sul Copyright	2
1.2	Registro delle modifiche	2
1.3	Ultima versione	3
2	Concetti generali	3
3	I programmi	3
3.1	apt-get	4
3.1.1	update	4
3.1.2	install	5
3.1.3	upgrade	5
3.1.4	dist-upgrade	6
3.1.5	remove	6
3.1.6	autoclean	6
3.1.7	clean	6
3.1.8	build-dep	7
3.1.9	source	7
3.2	apt-cache	7
3.2.1	search	7
3.2.2	show	7
3.3	dpkg	8
3.3.1	Questo file in che pacchetto si trova ?	8
3.4	dpkg-reconfigure	9
3.5	deborphan	9
3.6	debfooster	10
4	Interfacce al sistema APT	10
4.1	Modalità testo	10
4.1.1	aptitude	10
4.1.2	dselect	10
4.2	Modalità grafica	11
4.2.1	gnome-apt	11
4.2.2	kpackage	11
4.2.3	synaptic	11

5	I file di configurazione	11
5.1	/etc/apt/sources.list	12
5.1.1	I mirror di Debian	12
5.1.2	Aggiornamenti di sicurezza	12
5.1.3	Proposed updates	12
5.1.4	apt-get.org	13
6	I pacchetti più importanti	13
6.1	CUPS	13
6.2	XFree86	13
6.3	PPP over Ethernet	14
7	Compilare il kernel con Debian	14

1 Introduzione

Questo documento vuole essere una guida all'uso di APT per gestire i pacchetti di una distribuzione Debian. È pensato per crescere in futuro, fino a diventare una guida il più completa possibile, comprendendo anche tutte le possibilità che il sistema offre per la gestione dei pacchetti sorgente. Oltre ad APT viene anche spiegato l'uso di altri programmi di contorno che semplificano certe operazioni.

Questa guida è scritta in particolar modo per gli utenti alle prime armi, quelli più smaliziati al massimo possono trovarci qualche scorciatoia che prima non conoscevano.

1.1 Informazioni sul Copyright

Siccome questo documento è distribuito su Internet i termini di licenza e le informazioni di copyright sono in inglese. Sul sito Creative Commons

Copyright © Daniele Venzano, 2002, 2003. This document is licensed under the Creative Commons Attribution-NoDerivs-NonCommercial License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-nd-nc/1.0/> or send a letter to Creative Commons, 559 Nathan Abbott Way, Stanford, California 94305, USA.

1.2 Registro delle modifiche

- Versione 1.6 del 20 dicembre 2003:
 - Descrizione di kernel-package
 - Cambiamento di charset
- Versione 1.5 del 13 dicembre 2003:
 - Correzioni minori
 - Spiegazione apt-get build-dep
 - Spiegazione apt-get source
- Versione 1.4 del 3 ottobre 2003:
 - Cambio di licenza

- Versione 1.3 del 19 luglio 2003:
 - Sezione sui proposed updates
- Versione 1.2 del 16 febbraio 2003:
 - Aggiunta la descrizione di synaptic
 - Esempi per dpkg -S e dpkg -L
 - Sezione sugli aggiornamenti di sicurezza
- Versione 1.1 del 9 febbraio 2003:
 - Aggiunta la sezione sui pacchetti piú importanti
 - Aggiunta la descrizione di debfoster
 - Corretti un po' di errori di battitura

1.3 Ultima versione

L'ultima versione di questo documento è disponibile presso il sito dell'autore alla pagina:
<http://teg.homeunix.org/documentation.html>

2 Concetti generali

Debian nel corso degli anni ha sviluppato un sistema molto potente per la gestione dell'installazione/aggiornamento/rimozione dei pacchetti. Il sistema, che va sotto il nome di APT, è ancora in sviluppo da parte di un team di programmatori all'interno di Debian.

APT usa diverse sorgenti di pacchetti (cdrom, http, ftp, nfs, ...) per costruire una lista di tutti i pacchetti disponibili, con le loro dipendenze. La distribuzione stabile di Debian piú aggiornata ad oggi, chiamata Woody, è composta da 7 cd e contiene circa 10.000 pacchetti. Piú avanti vedremo come cercare un pacchetto che ci interessa, come installarlo, aggiornarlo e come cancellarlo quando ci ha stufato.

Il vantaggio che ha APT su altri sistemi simili, come RPM, è dato dalla lista di tutti i pacchetti disponibili che mantiene. Grazie a questa lista può risolvere tutte le dipendenze automaticamente al momento dell'installazione. Se gli si chiede di installare un gioco per Gnome, lui si costruirà una lista di quali altri pacchetti sono necessari per il buon funzionamento di quel gioco, chiedendo se deve procedere a installarli tutti.

L'estremo rigore con cui un nuovo pacchetto viene testato prima di essere inserito nell'archivio ufficiale garantisce che non ci siano mai dipendenze insoddisfabili a causa, ad esempio, di pacchetti che non esistono piú.

3 I programmi

La gestione dei pacchetti in Debian è composta principalmente da due programmi.

dpkg sa come è fatto un pacchetto .deb e si occupa di:

- Spacchettare il file .deb
- Eventualmente rimuovere i file di una versione precedente
- Copiare i nuovi file

- Porre all'utente una serie di domande di configurazione (tramite debconf)

apt-get si occupa invece di trovare e mettere a disposizione di dpkg i file .deb e quindi:

- Dato il nome di un pacchetto lo cerca nelle sue liste per scoprire da dove può prendere il file (ad es. server http o cdrom)
- Verifica che le dipendenze siano soddisfatte e nel caso aggiunge tutti i pacchetti necessari alla lista dei file da trovare
- Chiede conferma all'utente se necessario
- Si procura i file .deb (copiandoli da un CD o scaricandoli da Internet)
- Esegue dpkg passandogli tutti i pacchetti da installare

Piú in generale *apt-get* agisce da interfaccia per dpkg, semplificando i compiti di gestione delle dipendenze e occupandosi di trovare e copiare i pacchetti.

3.1 apt-get

apt-get è un programma da linea di comando e richiede la seguente sintassi:

```
apt-get {comando} [ opzioni ] [ nome pacchetto ]
```

Qui sotto sono elencati i comandi piú utili, con le opzioni relative. *man apt-get* contiene tutti gli altri comandi e le opzioni non elencate qui di seguito.

3.1.1 update

Sintassi:

```
apt-get update
```

update serve ad aggiornare le liste di pacchetti nel caso in cui le sorgenti puntino a un mirror Debian su Internet. Infatti se si utilizzano i cdrom i pacchetti non cambiano mai e non c'è bisogno di aggiornare le liste.

Nella versione stabile di Debian è possibile avere una sorgente di pacchetti che punta agli aggiornamenti di sicurezza, a questo modo si possono utilizzare i cdrom originali per tutti i pacchetti di cui non sia disponibile un aggiornamento, e solo questi ultimi vengono scaricati da Internet.

In una situazione come questa è bene aggiornare le liste di pacchetti almeno una volta alla settimana, è possibile impostare a mano un cron job per questo oppure utilizzare il pacchetto *cron-apt*.

Esempio di un aggiornamento delle liste di pacchetti:

```
myhost:~# apt-get update
Hit ftp://mi.mirror.garr.it testing/main Packages
Hit ftp://mi.mirror.garr.it testing/main Release
Hit ftp://ftp.tiscali.de testing/contrib Packages
Hit ftp://ftp.tiscali.de testing/contrib Release
Hit ftp://ftp-stud.fht-esslingen.de testing/non-free Packages
```

```
Hit ftp://ftp-stud.fht-esslingen.de testing/non-free Release
Reading Package Lists... Done
Building Dependency Tree... Done
myhost:~#
```

3.1.2 install

Sintassi:

```
apt-get install {nome pacchetto}
```

Install serve a installare un nuovo pacchetto o ad aggiornarne uno già installato. Il nome del pacchetto non deve contenere l'estensione n'è la versione.

Durante l'esecuzione del programma viene mostrata la dimensione dei pacchetti da scaricare rispetto al totale. Nel caso in cui delle dipendenze hanno causato l'aggiunta di altri pacchetti alla lista di installazione, viene anche chiesta conferma dell'operazione.

Esempio di installazione del pacchetto ksnapshot:

```
myhost:~# apt-get install ksnapshot
Reading Package Lists... Done
Building Dependency Tree... Done
The following NEW packages will be installed:
 ksnapshot
0 packages upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
Need to get 0B/56.5kB of archives. After unpacking 246kB will be used.
Selezionato il pacchetto ksnapshot, che non lo era.
(Lettura del database ... 106620 file e directory attualmente installati.)
Spacchettamento di ksnapshot (da ../ksnapshot_4%3a2.2.2-6.4_i386.deb) ...
Configuro ksnapshot (2.2.2-6.4) ...

myhost:~#
```

3.1.3 upgrade

Sintassi:

```
apt-get upgrade [ -u ]
```

Upgrade viene in genere utilizzato subito dopo update. Serve ad aggiornare tutti i pacchetti di cui sia disponibile una nuova versione. Questo comando però non tocca altri pacchetti, quindi se l'aggiornamento richiede l'installazione o la rimozione di altri pacchetti, viene mostrato un messaggio che elenca i pacchetti problematici e si procede solo con gli aggiornamenti che non causano modifiche al numero di pacchetti installati.

Come per install vengono mostrate alcune informazioni sulle dimensioni del pacchetti ed eventualmente viene chiesta la conferma prima di iniziare l'operazione.

L'opzione -u costringe apt-get a mostrare la lista dei pacchetti che stanno per essere aggiornati.

3.1.4 dist-upgrade

Sintassi:

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

Dist-upgrade funziona come upgrade, con la differenza che tutti i pacchetti vengono aggiornati, se c'è bisogno che altri pacchetti siano rimossi/installati, dist-upgrade aggiunge queste operazioni alla lista delle cose da fare, chiedendo conferma prima di iniziare.

3.1.5 remove

Sintassi:

```
apt-get remove [ -purge ] {nome pacchetto}
```

remove serve a rimuovere un pacchetto dal sistema. Normalmente i file di configurazione del programma che si sta rimuovendo non vengono cancellati. Questo può essere utile se si reinstalla il pacchetto in un secondo momento.

L'opzione `-purge` costringe apt-get a cancellare anche i file di configurazione.

Rimozione del pacchetto `ksnapshot` e dei suoi file di configurazione

```
myhost:~# apt-get remove --purge ksnapshot
Reading Package Lists... Done
Building Dependency Tree... Done
The following packages will be REMOVED:
 ksnapshot*
0 packages upgraded, 0 newly installed, 1 to remove and 0 not upgraded.
Need to get 0B of archives. After unpacking 246kB will be freed.
Do you want to continue? [Y/n]
(Lettura del database ... 106636 file e directory attualmente installati.)
Rimozione di ksnapshot ...
Eliminazione dei file di configurazione di ksnapshot ...
myhost:~#
```

3.1.6 autoclean

Sintassi:

```
apt-get autoclean
```

autoclean deve essere utilizzato quando la directory `/var/cache/apt/archives/` sta diventando troppo grande. Il comando cancella tutti i pacchetti scaricati da Internet di cui esistano delle nuove versioni.

3.1.7 clean

Sintassi:

```
apt-get clean
```

clean cancella tutti i file `.deb` presenti nella directory `/var/cache/apt/archives/`

3.1.8 build-dep

Sintassi:

```
apt-get build-dep {nome pacchetto}
```

build-dep scarica e installa tutti i pacchetti di sviluppo necessari a compilare un pacchetto.

3.1.9 source

Sintassi:

```
apt-get source [ -compile ] {nome pacchetto}
```

source scarica un pacchetto sorgente nella directory corrente e lo scompatta in una directory chiamata come il pacchetto. Se viene data l'opzione `-compile` il pacchetto viene anche compilato con le opzioni usate dal Maintainer e un `.deb` viene generato.

Il pacchetto deb può essere generato anche successivamente con `dpkg-buildpackage`, che deve essere lanciato nella directory che contiene i sorgenti.

Per funzionare, questo comando ha bisogno che ci sia almeno una sorgente di pacchetti sorgente in `/etc/apt/sources.list`

3.2 apt-cache

apt-cache serve a cercare o chiedere informazioni su un pacchetto.

3.2.1 search

Sintassi:

```
apt-cache search {chiavi di ricerca}
```

search ricerca tutti i pacchetti che contengono nella descrizione le chiavi di ricerca. Questo consente di cercare, ad esempio, tutti i pacchetti che riguardano html o il protocollo ftp.

Ricerca dei pacchetti che riguardano connessioni DSL su protocollo PPP:

```
user@myhost:~$ apt-cache search dsl ppp
dnsmasq - A caching DNS forwarder.
pppoe - PPP over Ethernet driver
pppoeconf - configures PPPoE/ADSL
pppstatus - console-based PPP status monitor
user@myhost:~$
```

3.2.2 show

Sintassi:

```
apt-cache show {nome pacchetto}
```

show mostra tutte le informazioni disponibili per un dato pacchetto. Queste informazioni comprendono la dimensione una volta installato, le dipendenze e una descrizione completa.

Descrizione del pacchetto pppoe:

```
user@myhost:~$ apt-cache show pppoe
Package: pppoe
Priority: optional
Section: net
Installed-Size: 150
Maintainer: Christian Hudon <chrish@debian.org>
Architecture: i386
Source: rp-pppoe
Version: 3.3-1.2
Depends: libc6 (>= 2.2.5-13), ppp (>= 2.3.10-1)
Filename: pool/main/r/rp-pppoe/pppoe_3.3-1.2_i386.deb
Size: 53820
MD5sum: acb821370c669f036251cced84308512
Description: PPP over Ethernet driver
 PPP over Ethernet (PPPoE) is a protocol used by
 many ADSL Internet service providers. This package allows
 you to connect to those PPPoE service providers.
Task: dialup

user@myhost:~$
```

3.3 dpkg

dpkg (Debian PacKaGe) viene raramente usato direttamente, molto piú spesso è richiamato da *apt-get* per lo spaccettamento/rimozione. È utile sapere come funziona perché ci sono dei casi in cui si ha già un file *.deb* che non si trova negli archivi Debian e che è stato scaricato senza usare *apt-get*.

Per installare un file *.deb* si usa l'opzione *-i*:

```
dpkg -i file.deb
```

Per disinstallare un pacchetto l'opzione è *-r*:

```
dpkg -r nome pacchetto
```

Per disinstallare un pacchetto cancellando anche i file di configurazione, l'opzione è *-P*:

```
dpkg -P {nome pacchetto}
```

3.3.1 Questo file in che pacchetto si trova ?

dpkg serve anche a rispondere a domande come quella qui sopra, in questo caso si usa l'opzione *-S*:

```
dpkg -S {nome file}
```

Come scoprire che il file */bin/ls* sta nel pacchetto *coreutils*:

```
user@myhost:~$ dpkg -S 'which ls'
coreutils: /bin/ls
user@myhost:~$
```

Invece per elencare tutti i file appartenenti a un pacchetto l'opzione è -L:

```
dpkg -L {nome pacchetto}
```

Come ottenere l'elenco dei file presenti nel pacchetto coreutils:

```
user@myhost:~$ dpkg -L coreutils
/.
/bin
/bin/cat
/bin/chgrp
/bin/chmod
/bin/chown
/bin/cp
...
/usr/sbin/chroot
/usr/share/man/man1/[.1.gz
/usr/bin/[
/usr/bin/touch
user@myhost:~$
```

3.4 dpkg-reconfigure

dpkg-reconfigure serve a riconfigurare un pacchetto già installato, ad esempio con *dpkg-reconfigure locales* si possono cambiare le impostazioni di ora/data/valuta/ecc. utilizzate dal sistema.

Quindi la sintassi è del tipo:

```
dpkg-reconfigure {nome pacchetto}
```

3.5 deborphan

deborphan è un programma a parte, che vive in un pacchetto a sè stante e serve a mantenere il sistema pulito da pacchetti che erano installati solo per soddisfare qualche dipendenza, ma che non sono stati rimossi quando non più necessari. Questo problema è particolarmente grave per i pacchetti che contengono librerie ed infatti *deborphan* di default cerca solo all'interno della sezione *libs*. Tramite l'opzione *-a* lo si può costringere a cercare tra tutti i pacchetti, ma c'è il rischio di vedere un sacco di falsi positivi. Ad esempio *xgalaga* è un gioco e nessun altro pacchetto dipende da lui per funzionare, per *deborphan* è un ottimo candidato all'eliminazione.

Per evitare questi risultati poco significativi, *deborphan* ha una serie di opzioni per cercare solo nelle sezioni in cui c'è la possibilità di trovare effettivamente qualcosa di inutile:

3.6 debfoster

debfoster cerca di fare le stesse cose di deborphan, ma in un modo diverso. Al primo avvio chiede quali tra i pacchetti installati sono da tenere. Poi, tenendo aggiornata questa lista in base alle aggiunte di nuovi pacchetti rileva quando una dipendenza che aveva provocato l'installazione di un pacchetto non è più necessaria. Nel caso chiede se si vuole rimuovere il pacchetto che non è più necessario.

4 Interfacce al sistema APT

4.1 Modalità testo

4.1.1 aptitude

aptitude è attualmente il front-end più facile da usare. Permette di fare tutte le operazioni che andrebbero fatte da linea di comando in maniera abbastanza semplice. Usa gli stessi comandi da tastiera di dselect.

Pro:

- Facile da usare
- Completamente configurabile
- Buona gestione delle dipendenze
- Dopo un update mostra i pacchetti nuovi

Contro:

- Lento a partire
- L'opzione per vedere i ChangeLog dei pacchetti non funziona

4.1.2 dselect

dselect è il front-end storico di Debian, è molto usato da chi ha imparato a convivere e ormai ci si trova a suo agio.

Pro:

-guess-dev	pacchetti di sviluppo
-guess-perl	moduli perl
-guess-section	pacchetti nella sezione sbagliata
-guess-debug	pacchetti con informazioni di debug
-guess-python	moduli python
-guess-pike	moduli pike
-guess-ruby	moduli ruby
-guess-interpreters	tutti i moduli di Perl, Python, Pike, Ruby
-guess-dummy	pacchetti che contengono solo dipendenze
-guess-common	pacchetti comuni a diverse implementazioni
-guess-all	tutte le opzioni di cui sopra

Tabella 1: Opzioni di deborphan

- Veloce da usare una volta che si è capito come funziona
- Completamente configurabile
- Un pezzo di storia di Debian

Contro:

- Difficile da imparare
- Noiosa la gestione delle dipendenze

4.2 Modalità grafica

I front-end grafici sono ancora molto indietro e quando sono stabili mancano di alcune funzionalità, ma lo sviluppo continua ed è bene darci un'occhiata di tanto in tanto per vedere i miglioramenti.

4.2.1 *gnome-apt*

gnome-apt è sviluppato in GTK appositamente per il sistema APT, è già utilizzabile per le funzioni base, ma serve ancora un po' di tempo prima che possa sperare di avvicinarsi alle possibilità di *dselect*.

4.2.2 *kpackage*

kpackage utilizza le librerie QT e l'ambiente KDE. È pensato per funzionare anche su distribuzioni basate su RPM ed è in uno stato più avanzato di *gnome-apt*. Per ora realizza solo le funzioni comuni a RPM e APT e non sono presenti alcune finezze implementate in un sistema, ma non nell'altro come la rimozione dei file di configurazione di APT.

4.2.3 *synaptic*

synaptic è molto stabile e facile da usare e ha un sistema di filtri che consente di vedere solo i pacchetti a cui si è interessati. Consente di gestire sia upgrade che dist-upgrade, inoltre usa una finestra apposita per l'output di *dpkg* che è utile per rivedere eventuali errori durante l'installazione dei pacchetti. Utilizza GTK per disegnare l'interfaccia grafica, ma è indipendente da Gnome. Ultimamente ha fatto dei passi in avanti e la versione attualmente disponibile in testing è perfettamente utilizzabile al posto di *dselect* o *aptitude*.

Ho trovato un solo difetto, finora: la finestra dove c'è la lista dei pacchetti è troppo corta e lascia vedere pochi pacchetti per volta.

5 I file di configurazione

APT ha bisogno di pochissima configurazione, l'unica cosa che c'è da modificare è la lista delle sorgenti dei pacchetti. È possibile avere anche un file di configurazione per i tanti parametri di APT, ma è raro sentirne il bisogno dato che tutte le opzioni hanno dei valori di default ottimali. Sul sito <http://www.apt-get.org> si possono trovare sorgenti aggiuntive per la versione stabile, in modo da avere le ultime versioni di programmi come Kde, Gnome e OpenOffice.

5.1 /etc/apt/sources.list

/etc/apt/sources.list contiene le informazioni che servono ad APT per trovare i pacchetti. Ogni riga definisce una sorgente di pacchetti, le righe che iniziano con un cancelletto (#) sono commenti e vengono ignorate. Esempio:

```
deb ftp://ftp.it.debian.org/debian/ testing main contrib non-free
```

deb: tipo di sorgente ftp://ftp.it.debian.org/debian/: URI della sorgente testing: nome della distribuzione dalla quale prendere i pacchetti main contrib non-free: lista di archivi

Varie alternative per ogni voce:

- Tipo di sorgente: *deb* per i pacchetti precompilati, *deb-src* per i pacchetti sorgente
- URI: Indirizzi ftp o http, cdrom (cdrom://) o directory (file://)
- Nome distribuzione: *stable*, *testing*, *unstable* oppure *potato*, *woody*, *sid*, ...
- Lista di archivi *main* per l'archivio principale, *contrib* per pacchetti molto usati ma che hanno licenze non GPL like o non del tutto chiare, *non-free* per pacchetti con licenze non compatibili con Debian Free Software Guidelines (DFSG)

5.1.1 I mirror di Debian

Una lista dei mirror di Debian in tutto il mondo si trova su <http://www.debian.org/mirror/list>, inoltre esistono un paio di script che aiutano a scegliere il mirror più veloce, *apt-spy* e *netselect*. Entrambi sono in grado di generare un file *sources.list* confrontando i tempi di latenza e la distanza in hop tra la macchina su cui girano e i diversi server.

Quasi tutte le URL fornite in questo manuale possono essere sostituite con quelle del vostro mirror preferito.

5.1.2 Aggiornamenti di sicurezza

Per la versione stabile di Debian è disponibile una sorgente per gli aggiornamenti di sicurezza, bug fix o nuove versioni. Per tutte le macchine collegate a Internet è possibile ottenere questi aggiornamenti aggiungendo questa riga al file /etc/apt/sources.list:

```
deb http://security.debian.org/ woody/updates main contrib non-free
```

5.1.3 Proposed updates

I *proposed updates* sono gli aggiornamenti proposti alla versione stabile di Debian per il rilascio della versione successiva. Spesso consentono di rendere una Debian stable po' più moderna. Per usufruirne bisogna aggiungere le seguenti sorgenti:

```
deb http://ftp.debian.org/debian proposed-updates main contrib non-free
```

5.1.4 apt-get.org

Il sito <http://www.apt-get.org> fornisce una lista di sorgenti per il sistema APT che consente di installare pacchetti del tutto assenti o più recenti di quelli forniti nell'archivio ufficiale. Molto spesso anziché di pacchetti creati ex-novo per una versione particolare di Debian, si tratta di *backport*, cioè di pacchetti presi dalla versione unstable e ricompilati per stable.

È da tener presente, però, che i pacchetti di apt-get.org non sono ufficiali e non sono garantiti gli aggiornamenti di sicurezza. Malgrado questo sono utilissimi per avere sempre l'ultima versione di Kde o Gnome.

6 I pacchetti più importanti

Questa sezione è basata sulla più recente distribuzione stabile di Debian, la Woody, che contiene più di 10.000 pacchetti. È utile avere un elenco di quali pacchetti bisogna installare per avere determinati servizi, anche se nella maggior parte dei casi il sistema di dipendenze provvede a installare tutto il necessario.

6.1 CUPS

Il Common Unix Printing System è un sistema di stampa, basato sul modello client/server, ed è configurabile con un'interfaccia web molto semplice. CUPS è composto dai seguenti pacchetti:

- cupsys: server
- cupsys-bsd: comandi del sistema di stampa BSD, lpr, lpq, lprm, ...
- cupsys-client: client
- cupsys-driver-gimpprint: driver per quasi tutte le stampanti in commercio

Una volta installato basta collegarsi con un browser a <http://localhost:631/> e usare l'utente 'root' quando richiesto per i compiti amministrativi.

Questo comportamento è modificabile dal file `/etc/cups/cupsd.conf`

6.2 XFree86

È possibile installare sia la versione 3.3.6 che la 4.1.0, quest'ultima può essere installata tramite il pacchetto `x-window-system-core` che ha dipendenze su tutti gli altri pacchetti necessari ad un sistema base. Poi ci sono altri pacchetti che possono essere utili:

- xlibs-dev: librerie statiche e header per compilare programmi che usano X
- xfonts-100dpi-transcoded: font aggiuntivi
- xfonts-75dpi-transcoded: font aggiuntivi
- xfonts-base-transcoded: font aggiuntivi
- xspecs: manuali e specifiche tecniche per programmare con X

Bisogna tenere presente che `x-window-system-core` non installa un window manager, quindi bisogna aggiungere il proprio programma preferito (Icewm, Enlightenment, Gnome, Kde, Ion, ...).

6.3 PPP over Ethernet

Il protocollo PPP over Ethernet è molto usato per i collegamenti a Internet tramite ADSL, in Debian sono forniti i pacchetti `pppoe` e `pppoeconf`. Quest'ultimo contiene un semplice programma di configurazione che rileva automaticamente il modem in uso e la scheda di rete a cui è collegato e genera tutti i file di configurazione necessari.

7 Compilare il kernel con Debian

Il pacchetto `kernel-package` consente di compilare un kernel e crearne un pacchetto Debian pronto da installare, come se fosse uno dei kernel distribuiti nell'archivio ufficiale.

L'albero che contiene i sorgenti del kernel dovrà già essere stato configurato con il solito `make menuconfig` (o con il vostro front end preferito). Poi con il comando `make-kpkg clean` si cancelleranno tutti i file rimasti indietro da una vecchia compilazione.

Infine per compilare il kernel e i moduli e per preparare il pacchetto Debian basta dare il comando:

```
fakeroot make-kpkg --revision=tiny.1.0 kernel_image'
```

Osservazioni:

- Al posto di `fakeroot` si può usare `sudo` o eseguire il comando direttamente da `root`.
- L'argomento dell'opzione `--revision` serve a distinguere il pacchetto creato da quelli presenti nell'archivio Debian, è consigliato l'utilizzo di un numero di versione che sia incrementato per ogni nuovo pacchetto, in modo da creare problemi al momento dell'aggiornamento.