La guida Debian rapida

Osamu Aoki <osamu\#at\#debian.org>
Traduzione italiana: Davide Di Lazzaro <mc0315\#at\#mclink.it>
'Autori' a pagina 23

CVS, gio gen 18 11:54:32 UTC 2007

Estratto

La guida Debian rapida (http://qref.sourceforge.net/) è pensata per fornire un breve introduzione al sistema Debian come **riferimento rapido**. Essa è un riassunto de La guida Debian (http://qref.sourceforge.net/).

Avviso di Copyright

Indice

Capitolo 1

Prefazione

Questo documento è nato come "guida rapida", poi è cresciuto. Nondimeno, **Keep it short and simple (Sii semplice e conciso)** (KISS) è il mio principio guida.

1.1 Convenzioni utilizzate

"La guida Debian rapida" fornisce informazioni attraverso brevi comandi `bash`.

Il riferimento a:

- una **pagina di manuale** in stile UNIX è dato sotto forma di `bash(1)`.
- una **pagina TEXINFO** GNU è dato sotto forma di `info libc`.

1.2 Le basi delle distribuzioni Debian

Debian mantiene tre distribuzioni diverse simultaneamente. Esse sono:

- `stable` — Ottima per organizzare un server, dato che è l'unica a ricevere aggiornamenti per la sicurezza.
- `testing` — la distribuzione preferita per una workstation, siccome contiene le ultime versioni di software per desktop che ha ricevuto un minimo di prove.
- `unstable` — Sul filo del rasoi. la scelta degli sviluppatori Debian

Quando i pacchetti in `unstable` non hanno annunci di bachi release-critical (RC) da almeno una settimana o più, vengono automaticamente passati a `testing`.

Le distribuzioni Debian hanno anche dei nomi in codice. Prima del rilascio di Woody, nell'agosto del 2002, le tre distribuzioni erano, rispettivamente, Potato, Woody, e Sid. Dopo il rilascio di Woody erano diventate, Woody, Sarge, e Sid. Quando verrà rilasciata Sarge, le distribuzioni `stable` ed `unstable` diventeranno Sarge e Sid; verrà quindi creata una nuova `testing` (all'inizio come copia di `stable`) e le verrà dato un nuovo nome in codice.

Sottoscrivete alla lista di messaggi a basso volume `debian-devel-announce@lists.debian.org` per annunci importanti su Debian.

Se volete usare dei pacchetti con versioni più recenti di quelle rilasciate con la distribuzione che state usando, allora o aggiornate ad una distribuzione più nuova, come descritto in 'Aggiornare una distribuzione a `stable`, `testing` o `unstable`' nella pagina successiva, oppure aggiornate solo i singoli pacchetti. Se questi non possono essere aggiornati facilmente, allora potreste pensare ad un backport, come descritto in 'Portare un pacchetto nel sistema `stable`' a pagina 18.

Capitolo 2

Aggiornare una distribuzione a `stable`, `testing` o `unstable`

2.1 Aggiornare da Potato a Woody

Questa procedura separatamente perchè l'APT di Potato non aveva tutte le funzioni descritte nella pagina di manuale attuale. `apt_preferences(5)`.

Dopo aver inserito i puntatori per Woody in `/etc/apt/sources.list`, aggiornate APT ed i pacchetti fondamentali alle versioni di Woody facendo quanto segue.

```
# apt-get update
# apt-get install libc6 perl libdb2 debconf
# apt-get install apt apt-utils dselect dpkg
```

Poi aggiornate il resto del sistema a Woody.

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

2.2 Preparazione per l'aggiornamento

Potete eseguire l'aggiornamento da una distribuzione all'altra prendendo i pacchetti dalla rete. Ciò può essere fatto come segue.

Procuratevi una lista pulita di depositi per `stable`:

```
# cd /etc/apt
# cp -f sources.list sources.list.old
# :>sources.list
# apt-setup noprobe
```

Se volete aggiornare a `testing`, allora aggiungete i riferimenti a `testing` a questa nuova lista. Se volete farlo a `unstable`, aggiungete i riferimenti ad `unstable`.

```
# cd /etc/apt
# grep -e "^deb " sources.list  >srcs
# :>sources.list
# cp -f srcs sources.list
# sed -e "s/stable/testing/" srcs >>sources.list
# sed -e "s/stable/unstable/" srcs >>sources.list
# apt-get update
# apt-get install apt apt-utils
```

Vedere 'Iniziare la gestione dei pacchetti Debian' a pagina 6 per l'arte della messa a punto di `/etc/apt/sources.list` e `/etc/apt/preferences`.

2.3 Aggiornamento

Dopo aver impostato appropriatamente `/etc/apt/sources.list` e `/etc/apt/preferences` come descritto sopra, potete iniziare l'aggiornamento.

Notate che il tracciamento della distribuzione `testing` di Debian può avere come effetto collaterale il ritardo nell'installazione dei pacchetti contenenti correzioni per la sicurezza, poicè detti pacchetti vengono caricati su `unstable` e solo dopo un lasso di tempo migrano in `testing`.

Vedere 'Gestione dei pacchetti in Debian' a fronte per le basi e 'Aggiornamento con APT: risoluzione dei problemi' a pagina 11 se incontrate problemi.

2.3.1 Usare `dselect`

Se un sistema ha molti pacchetti, inclusi i pacchetti `-dev`, ecc., il metodo seguente tramite `dselect` è il metodo di scelta per un più raffinato controllo sui pacchetti.

```
# dselect update  # da fare sempre prima di un aggiornamento
# dselect select  # sceglie i pacchetti addizionali
```

Quando `dselect` parte, tutti i pacchetti correnti verranno selezionati. `dselect` potrà mostrarvi pacchetti addizionali basati su `Dipende`, `Suggerisce`, e `Raccomanda`. Se non volete altri pacchetti, basta pigiare `Q` per uscire di nuovo da `dselect`.

```
# dselect install
```

Dovrete rispondere ad alcune domande sulla configurazione dei pacchetti durante questa perte del processo, per cui tenete con voi i vostri appunti e prendetevi del tempo per questa parte. Vedere '`dselect`' a pagina 8.

Usate `dselect`. **Funziona sempre :)**

Capitolo 3

Gestione dei pacchetti in Debian

`aptitude` è attualmente l'interfaccia preferita per APT, l'Advanced Package Tool. Ricorda quali pacchetti avete deliberatamente installato e quali in base alle dipendenze; questi ultimi vengono disinstallati automaticamente da `aptitude` queo non sono più necessari ai pacchetti installati deliberatamente. Ha funzionalità avanzate di filtro sui pacchetti, ma possono essere difficili da configurare.

`synaptic` è attualmente l'interfaccia grafica Gtk preferita per APT. Le sue funzionalità di filtro sono più facili da usare rispetto ad `aptitude`. Ha anche il supporto sperimentale per le Debian Package Tags (`http://debtags.alioth.debian.org/`).

Per ridurre il carico sui depositi Debian e per velocizzare i vostri download, dovreste prendere i pacchetti dai mirror Debian.

Se dovete installare lo stesso pacchetto su più macchine sulla vostra rete locale, impostate un proxy HTTP locale useo `squid` per i pacchetti scaricati con APT. Impostate, se necessario, la variabile d'ambiente `http_proxy`, oppure il valore di `http` in `/etc/apt/apt.conf`.

Sebbene l'a funzionalità di pinning di APT, descritta in `apt_preferences(5)` sia potente, i suoi effetti possono essere difficili da comprendere e gestire. Consideratela una Funzione Avanzata.

L'uso di chroot è preferibile per assicurare al contempo stabilità del sistema ed accesso alle versioni più recenti del software.

Questo capitolo fa riferimento ad un sistema successivo a Woody Alcune funzionalità potrebbero richiedere un sistema Sarge o successivo.

3.1 Introduzione

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

3.1.1 Strumenti principali di gestione dei pacchetti

```
dpkg      - Installatore dei pacchetti Debian
apt-get   - Interfaccia a riga di comeo per APT
aptitude  - Interfaccia avanzata testo e riga di comeo per APT
synaptic  - Interfaccia grafica Gtk di APT
dselect   - installatore Debian dei pacchetti
tasksel   - Installatore dei task
```

Questi strumenti non non sono tutti alternativi fra loro. Per esempio, `dselect` usa sia APT che `dpkg`.

APT usa `/var/lib/apt/lists/*` per tenere traccia dei pacchetti disponibili, mentre `dpkg` usa `/var/lib/dpkg/available`. Se avete installato dei pacchetti useo direttamente `aptitude` od un'altra interfaccia ad APT, e volete usare `dselect` per installare dei pacchetti, assicuratevi di aggiornare il file `/var/lib/dpkg/available` tramite la voce `[A]ggiorna` del menu di `dselect` (o con il comeo "`dselect update`").

`apt-get` installa automaticamente tutti i pacchetti dai quali un pacchetto richiesto Dipende. Non installa i pacchetti che Raccomea o Suggerisce.

`aptitude`, al contrario, può essere configurato per installare i pacchetti che un pacchetto richiesto Raccomea o Suggerisce.

`dselect` presenta all'utente una lista di pacchetti che un pacchetto richiesto Raccomea o Suggerisce e permette una loro selezione caso per caso.

3.1.2 Strumenti utili

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

3.2 Iniziare la gestione dei pacchetti Debian

3.2.1 Impostare APT

Impostate `sources.list` come descritto in 'Preparazione per l'aggiornamento' a pagina 3. [1]

[1] Se state traccieo `testing` o `unstable`, potete rimuovere i riferimenti a `stable` da `/etc/apt/sources.list` e da `/etc/apt/preferences`. Ciò perchè `testing` origina come una copia di `stable`.

3.2.2 Installare i task

E' possibile installare un gruppo di pacchetti tipicamente richiesto per un determinato utilizzo del sistema Debian. Questo gruppo di pacchetti è chiamato "task".

Il modo più semplice per installare i task al momento dell'installazione iniziale è useo `tasksel`. Notate che dovete eseguire

```
dselect update
```

prima di utilizzarlo.

Anche `aptitude` può installare i task ed è lo strumento racomeato per questo scopo. Permette di deselzionare singoli pacchetti all'interno dei task prima di procedere all'installazione.

3.2.3 `aptitude`

`aptitude` è un nuovo installatore di pacchetti basato su menu, simile a `dselect`, ma costruito da zero sopra APT. Può essere utilizzato in alternativa ad `apt-get` per gran parte dei comandi. Vedere `aptitude(1)` e `/usr/share/doc/aptitude/README`.

Una volta cominciato ad usare `aptitude`, è meglio continuare ad usarlo invece che come metodo alternativo di installazione dei pacchetti; altrimenti perdete il vantaggio di avere `aptitude` che mantiene una traccia di tutti i pacchetti installati deliberatamente.

`aptitude` in modalità a tutto schermo accetta comandi sotto forma singole lettere, di norma in minuscolo. Tasti degni di nota sono:

```
Tasto       Azione
F10         Menu
?           Aiuto per i comandi
u           Aggiorna le informazioni relative all'archivio dei pacchetti
+           Marca il pacchetto nuovo da installare o da aggiornare
-           Marca il pacchetto da rimuovere (mantiene la configurazione)
_           Marca il pacchetto da eliminare (rimuove la configurazione)
=           Pone il pacchetto in attesa
U           Marca tutti i pacchetti aggiornabili da aggiornare
g           Scarica ed installa i pacchetti selezionati
q           Esce dalla schermata corrente e salva le modifiche
x           Esce dalla schermata corrente e non salva le modifiche
Enter       Mostra le informazioni su un pacchetto
C           Mostra il changelog di un pacchetto
l           Cambia i limiti per il pacchetto mostrato
/           Cerca la prima corrispondenza
\           Ripete l'ultima ricerca
```

Come `apt-get`, `aptitude` installa i pacchetti da cui un pacchetto selezionato Dipende. `aptitude` offre anche l'opzione di prendere tutti i pacchetti che un dato pacchetto Raccomea o Suggerisce. Potete modificare questo comportamento tramite `F10 -> Options -> Dependency heling` nel suo menu.

Altri vantaggi di `aptitude` sono:

- `aptitude` offre l'accesso a tutte le versioni di un pacchetto.
- `aptitude` registra le azioni in `/var/log/aptitude`.
- `aptitude` facilita il tenere traccia di software obsoleto elenceolo in "Obsolete e Locally Created Packages".
- `aptitude` include un sistema abbastanza potente per la ricerca di pacchetti particolari e per limitare i pacchetti mostrati. Gli utenti familiari con `mutt` ci si abitueranno rapidamente, poichè mutt è stata l'ispirazione per la sintassi. Vedere "SEARCHING, LIMITING, e EXPRESSIONS" in `/usr/share/doc/aptitude/README`.
- `aptitude` nella modalità a tutto schermo ha la funzione di `su` incastonata e può essere fatto girare da normale utente finchè non si ha realmente bisogno dei privilegi amministrativi.

3.2.4 `dselect`

Nelle versioni stabili da Potato in su, `dselect` era il principale strumento di gestione dei pacchetti. Per Sarge, considerate invece di usare `aptitude`.

Una volta lanciato, `dselect` seleziona automaticamente tutti i pacchetti "Essenziali", "Importanti", e "Steard".

`dselect` presenta un'interfaccia utente un pò strana. Comunque sia, sono in tanti ad essersi abituati. Quattro sono i comandi ambigui (Maiuscolo significa MAIUSCOLO!):

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

Useo `D` e `Q`, potete scegliere delle selezioni che presentano conflitti di dipendenze a vostro rischio e pericolo. Usate questi comand con cura.

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

Se la vostra macchine è lenta nell'esecuzione di `dselect`, prendete in considerazione di eseguire `dselect` su un'altra macchina (più veloce) per trovare i pacchetti che vi servono, poi usate `apt-get install` sulla macchina lenta per installarli.

3.2.5 Tracciare una distribuzione con APT

Per tracciare la distribuzione `testing` man mano che cambia, modificate il vostro `/etc/apt/preferences` in maniera che somigli a quanto segue:

```
Package: *
Pin: release a=testing
Pin-Priority: 800

Package: *
Pin: release a=stable
Pin-Priority: 600
```

Notate che il tracciamento della distribuzione `testing` può avere come effetto il ritardo nell'installazione degli aggiornamenti per la sicurezza. Tali pacchetti vengono prima caricati in `unstable` e poi migrano in `testing` solo dopo un lasso di tempo.

Vedere `apt_preferences(5)` per esempi più complessi che permettono, per esempio, di tracciare `testing` mentre si installano pacchetti selezionati da `unstable`.

Esempi di che bloccano alcuni pacchetti a determinate versioni, mentre si tracciano altri pacchetti via via che vengono rilasciati, sono disponibili nella sottodirectory degli esempi (`http://www.debian.org/doc/manuals/debian-reference/examples/`) come `preferences.testing` e `preferences.unstable`.

Se mischiate le distribuzioni, p.es. `testing` con `stable` o `unstable` con `stable`, vi troverete infine a prendere pacchetti fondamentali, come `libc6` da `testing` o `unstable` e non c'è garanzia che essi non conterranno bachi. Siete stati avvertiti.

Un altro esempio, `preferences.stable`, forza tutti i pacchetti ad essere degradati a `stable`.

Il declassamento di un **pacchetto** da una versione più recente ad una meno non è ufficialmente supportato in Debian. Tuttavia, potreste aver bisogno di declassare un dato pacchetto per reinstallarne una versione che sia funzionante, queo quella più nuova non lo è. Queste versione precedenti le trovate localmente in `/var/cache/apt/archives/` o remotamente in `http://snapshot.debian.net/`. Vedere anche 'Salvataggio mediante `dpkg`' a pagina 12.

Anche il declassamento di una **distribuzione** ad una meno recente non è ufficialmente supportato e quasi sicuramente causerà problemi. Tuttavia, se siete disperati, può valer la pena tentarlo come ultima risorsa.

3.2.6 I comandi di `aptitude`, `apt-get` e `apt-cache`

Mentre tracciate `testing`, come descritto sopra, potete gestire il sistema con i comandi seguenti:

- `aptitude upgrade` (o `apt-get upgrade` o `aptitude dist-upgrade` o `apt-get dist-upgrade`)

  Tracciano la distribuzione `testing` — aggiornano tutti i pacchetti del sistema dopo aver installato le versioni dei pacchetti da cui Dipende, dalla distribuzione `testing`. [2]

- `apt-get dselect-upgrade`

  Traccia `testing` — aggiorna tutti i pacchetti in accordo con la selezioni di `dselect`.

- `aptitude install` *`pacchetto`*`/unstable`

  Installa *pacchetto* da `unstable`, mentre installa le sue dipendenze da `testing`.

- `aptitude install -t unstable` *`pacchetto`*

  Installa *pacchetto* da `unstable`, mentre installa le sue dipendenze anche da `unstable` impostando la Pin-Priority di `unstable` a 990.

- `apt-cache policy` *`foo bar ...`*

  Controlla lo stato dei pacchetti *foo bar* ....

- `aptitude show` *`foo bar ...`* `| less` (o `apt-cache show` *`foo bar ...`* `| less`)

  Controlla le informazioni per i pacchetti *foo bar* ....

- `aptitude install` *`foo=2.2.4-1`*

  Installa la data versione *2.2.4-1* del pacchetto *foo*.

- `aptitude install` *`foo bar-`*

  Installa il pacchetto *foo* e rimuove *bar*

- `aptitude remove` *`bar`*

  Rimuove il pacchetto *bar* ma non i suoi file di configurazione.

- `aptitude purge` *`bar`*

  Rimuove *bar* insieme ai suoi file di configurazione.

Nell'esempio sopra, dando ad `apt-get` l'opzione `-u` gli fa scrivere una lista di tutti i pacchetti da aggiornare e aspettare l'utente prima di intraprendere le azioni. Quanto segue gli fa sempre `apt-get` fare così:

```
$ cat >> /etc/apt/apt.conf << .
// Mostra sempre i pacchetti che devono essere aggiornati (-u)
APT::Get::Show-Upgraded "true";
.
```

[2] La differenza fra `upgrade` e `dist-upgrade` è evidente solo queo nuove versioni dei pacchetti hanno relazioni di dipendenza diverse rispetto alle vecchie versioni. Vedere `apt-get(8)` per i dettagli. `aptitude upgrade` e `aptitude dist-upgrade` lanciano `aptitude` in modalità riga di comando. Potete passare a tutto schermo premendo il tasto `e`.

Utilizzare l'opzione `--no-act` per simulare le azioni senza realmente installare, rimuovere, ecc. alcun pacchetto.

3.3 Comandi di sopravvivenza Debian

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

3.3.1 Controllo dei bachi in Debian e ricerca di aiuto

Se avete dei problemi con un pacchetto in particolare, controllate prima questi siti e poi, eventualmente, cercate aiuto o segnalate un baco. (`lynx`, `links` e `w3m` funzionano tutti bene):

```
$ lynx http://bugs.debian.org/
$ lynx http://bugs.debian.org/nome-pacchetto  # se conoscete il nome del pacchetto
$ lynx http://bugs.debian.org/numerobaco     # se conoscete il numero del baco segnalato
```

Cercate su Google (www.google.com) includendo nelle frasi da ricercare "site:debian.org".

Se avete ancora dubbi, leggete i manuali. Impostate `CDPATH` come segue:

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

e digitate

```
$ cd nomepachetto
$ pager README.Debian # se esiste
$ mc
```

3.3.2 Aggiornamento con APT: risoluzione dei problemi

Problemi di dipendenze possono accadere quando si esegue l'aggiornamento in `unstable` o `testing`, come descritto in 'Aggiornamento' a pagina 4. Molte volte ciò accade poichè un pacchetto da aggiornare presenta delle nuove dipendenze che non vengono soddisfatte. Problemi del genere vengono risolti usando

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

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

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

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

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

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

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

```
# aptitude -f install pacchetto # ignora le dipendenze difettose
```

Un metodo alternativo consiste nell'utilizzare il pacchetto `equivs`. Vedere `/usr/share/doc/equivs/README.Debian`.

3.3.3 Salvataggio mediante dpkg

Se siete in un vicolo cieco con APT, potete scaricare i pacchetti dai mirror Debian ed installarli con `dpkg`. Se non avete accesso alla rete, potete cercare le copie dei pacchetti in `/var/cache/apt/archives/`.

```
# dpkg -i fetchmail_6.2.5-4_i386.deb
```

Se l'installazione di un pacchetto fallisce a causa di violazioni delle dipendenze e avete realmente bisogno di quel pacchetto, potete scavalcare i controlli sulle dipedenze usando le opzioni di `dpkg` `--ignore-depends`, `--force-depends` ed altre ancora. Vedere `dpkg(8)` per i dettagli.

3.3.4 Recuperare i dati sulla selezione dei pacchetti

Se `/var/lib/dpkg/status` diventa illeggibile per una qualsivoglia ragione, il sistema Debian perde i dati sulla selezione dei pacchetti, soffrendone in maniera severa. Cercate il vecchio file `/var/lib/dpkg/status-old`, oppure `/var/backups/dpkg.status.*`.

Tenere `/var/backups/` in una partizione separata può essere una buona idea, dato che questa directory contiene molti dei dati fondamentali del sistema.

Se il vecchio file `/var/lib/dpkg/status` non è disponibile, potete ancora recuperare le informazioni dalle directory in `/usr/share/doc/`.

```
# ls /usr/share/doc | \
  grep -v [A-Z] | \
  grep -v '^texmf$' | \
  grep -v '^debian$' | \
  awk '{print $1 " install"}' | \
  dpkg --set-selections
# dselect --expert # reinstalla il sistema, deseleziona se necessario
```

3.3.5 Recupero del sistema dopo danneggiamento di /var

Dato che la directory /var contiene dati che vengono regolarmente aggiornati, tipo la posta, è più suscettibile di corruzione di, p. es., /usr/ Metterla in una partizione separata limita i rischi. Se accade l'irreparabile, bisogna ricostruirla per salvare il sistema Debian.

Procuratevi una directory /var con il contenuto ridotto all'osso, da un sistema Debian minimale e funzionante, basato su una versione uguale o più vecchia, per esempio var.tar.gz(http://people.debian.org/~osamu/pub/) e piazzatela nella root directory del sistema danneggiato. Poi,

```
# cd /
# mv var var-old      # se vi è rimasto qualcosa di utile
# tar xvzf var.tar.gz # usate il file preso da Woody
# aptitude            # o dselect
```

Ciò dovrebbe restituirvi un sistema funzionante. Potete velocizzare la procedura di recupero delle selezioni dei pacchetti usando la tecnica descritta in 'Recuperare i dati sulla selezione dei pacchetti' a fronte. ([FIXME]: This procedure needs more experiments to verify.)

3.3.6 Installare un pacchetto su un sistema non bootabile

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

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

Poi configurate il tutto e risolvete i problemi.

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

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

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

3.3.7 Cosa fare se dpkg non funziona

Un dpkg malfunzionante può rendere impossibile l'installazione di qualsiasi file .deb. Una procedura come la seguente vi aiuterà ad uscire da questa situazione. (Nella prima riga, potete sostituire "links" con il vostro browser da terminale preferito.)

```
$ links http://http.us.debian.org/debian/pool/main/d/dpkg/
... scarica un valido dpkg_version_arch.deb
$ su
password: *****
$ ar x dpkg_version_arch.deb
# mv data.tar.gz /data.tar.gz
# cd /
# tar xzfv data.tar.gz
```

Per i386, http://packages.debian.org/dpkg può anche essere utilizzata come URL.

3.4 Comandi del "paradiso" Debian

L'**Illuminazione** ottenuta con questi comandi salverà una persona dall'eterno ciclo del karma dell'inferno degli aggiornamenti, permettendogli di raggiungere il **nirvana** Debian. :-)

3.4.1 Informazioni su di un file

Per scoprire a quale pacchetto un file appartiene:

```
$ dpkg {-S|--search} modello # trova i modelli nei pacchetti installati
$ wget http://ftp.us.debian.org/debian/dists/sarge/Contents-i386.gz
$ zgrep -e pattern Contents-i386.gz
      # trova l'elenco dei file con la stessa radice (o il file) nell'archivio Debian
```

Potete utilizzare anche dei comandi specifici per i pacchetti:

```
# aptitude install dlocate
$ dlocate filename # alternativa rapida a dpkg -L e dpkg -S
...
# aptitude install auto-apt # strumento per l'installazione a richiesta dei pacchetti
# auto-apt update          # crea un database per auto-apt
$ auto-apt search modello
 # effettua una ricerca completa nell'archivio pacchetti
```

3.4.2 Informazioni su di un pacchetto

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

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

`apt-cache showsrc` non è documentato al momento del rilascio di Woody, ma funziona :)

E' sempre possibile trovare informazioni sui pacchetti in (Io uso `mc`):

```
/var/lib/apt/lists/*
/var/lib/dpkg/available
```

Il confronto dei file seguenti fornisce informazioni su cosa è realmente accaduto nelle ultime sessioni di installazione.

```
/var/lib/dpkg/status
/var/backups/dpkg.status*
```

3.4.3 Installazione automatica con APT

Per una installazione automatica, senza controllo alcuno, aggiungete la seguente riga a `/etc/apt/apt.conf`:

```
Dpkg::Options {"--force-confold";}
```

Questo è l'equivalente di `aptitude -y install` *nomepacchetto*. o `apt-get -q-y install` *nomepacchetto*. Siccome in questo modo si risponde "si" a tutte le domande, usate questo trucco con accortezza. Vedere `apt.conf(5)` e `dpkg(1)`.

Si può configurare qualsiasi pacchetto in un secondo momento seguendo 'Riconfigurare i pacchetti già installati' nella pagina successiva.

3.4.4 Riconfigurare i pacchetti già installati

Usate i seguenti comandi per riconfigurare un pacchetto già installato.

```
# dpkg-reconfigure --priority=medium pacchetto [...]
# dpkg-reconfigure --all  # riconfigura tutti i pacchetti
# dpkg-reconfigure locales # genera qualsiasi locale extra
# dpkg-reconfigure --p=low xserver-xfree86 # riconfigura il server X
```

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

Alcuni programmi hanno degli script speciali di configurazione. [3]

```
apt-setup     - crea /etc/apt/sources.list
install-mbr   - installa un manager di Master Boot Record
tzconfig      - imposta il fuso orario locale
gpmconfig     - imposta il demone per mouse gpm
eximconfig    - configura Exim (MTA)
texconfig     - configura teTeX
apacheconfig  - configura Apache (httpd)
cvsconfig     - configura CVS
sndconfig     - configura il sistema audio
...
update-alternatives - imposta i comandi predefiniti, p.e. vim per vi
update-rc.d         - Gstione degli script di inzializzazione del System-V
update-menus        - Il sistema dei menu Debian
...
```

3.4.5 Rimozione e purga dei pacchetti

Rimuove un pacchetto mantenendone la configurazione:

```
# aptitude remove pacchetto ...
# dpkg   --remove pacchetto ...
```

Rimuove un pacchetto e la sua configurazione:

```
# aptitude purge pacchetto ...
# dpkg   --purge       pacchetto ...
```

[3] Alcuni script `*config` stanno scomparendo nelle nuove versioni di Sarge e la funzionalità di configurazione dei pacchetti viene trasferita al sistema `debconf`.

3.4.6 Mantenere vecchi pacchetti

Per esempio, per mantenere `libc6` e `libc6-dev` con `dselect` e `aptitude install` *`pacchetto`* si può procedere come segue:

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

`aptitude install` *`pacchetto`* non verrà impedito da quell'"hold". Per mantenere un pacchetto forzandone un declassamento automatico con `aptitude upgrade` *`pacchetto`* oppure `aptitude dist-upgrade`, aggiungete le seguenti linee ad `/etc/apt/preferences`:

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

Qui la voce "`Package:`" non può usare termini tipo "`libc6*`". Se dovete mantenere tutti i pacchetti binari correlati al pacchetto sorgente `glibc` in una versione sincronizzata, dovete elencarli esplicitamente.

Il comando seguente mostra i pacchetti tenuti in sospeso (non aggiornati):

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

3.4.7 Sistema misto `stable/testing/unstable`

`apt-show-versions` può elencare le versioni dei pacchetti disponibili per ciascuna distribuzione.

```
$ apt-show-versions | fgrep /testing | wc
... quanti pacchetti avete da testing
$ apt-show-versions -u
... elenco di pacchetti aggiornabili
$ aptitude install `apt-show-versions -u -b | fgrep /unstable`
... aggiorna tutti i pacchetti da unstable alle loro versioni più recenti
```

3.4.8 Potare i file della cache

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

```
# aptitude autoclean # rimuove solo pacchetti inutili
# aptitude clean     # rimuove tutti i pacchetti nella cache
```

3.4.9 Salvare/copiare la configurazione del sistema

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

```
# debconf-get-selections > debconfsel.txt
# dpkg --get-selections "*" >lemieselezioni   # oppure usare \*
```

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

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

```
# dselect update
# debconf-set-selections < debconfsel.txt
# dpkg --set-selections <lemieselezioni
# apt-get -u dselect-upgrade # o dselect install
```

3.4.10 Portare un pacchetto nel sistema `stable`

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

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

Ogni linea che inizia con `deb-src` qui è divisa in due parti per ragioni di stampabilità, ma in `sources.list` dovranno essere ciascuna su un'unica riga.

Poi recuperate i sorgenti e costruite un pacchetto locale:

```
$ apt-get update  # aggiorna la lista dei pacchetti sorgente
$ apt-get source pacchetto
$ dpkg-source -x pacchetto.dsc
$ cd versione-pacchetto
  ... controlla i pacchetti necessari (Build-depends nel file
      .dsc) e li installa. Avete bisogno anche del pacchetto "fakeroot".

$ dpkg-buildpackage -rfakeroot

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

```
...poi per installare il pacchetto
$ su -c "dpkg -i pacchetto.deb"
```

Normalmente, per soddisfare le dipendenze si ha la necessità di installare alcuni pacchetti con il suffisso "-dev". `debsign` è contenuto nel pacchetto `devscripts`. `auto-apt` può aiutare nel soddisfare queste dipendenze. `fakeroot` evita l'uso non necessario del root account.

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

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

3.4.11 Archivio locale dei pacchetti

Per creare un archivio locale dei pacchetti che sia compatibile con APT e `dselect`, bisogna creare `Packages` ed i file dei pacchetti devono essere organizzati in un particolare albero directory.

Si può creare un deposito locale `deb` simile ad un archivio ufficiale Debian in questo modo:

```
# aptitude install dpkg-dev
# cd /usr/local
# install -d pool # i pacchetti sono fisicamente localizzati qui
# install -d dists/unstable/main/binary-i386
# ls -1 pool | sed 's/_.*$/ priority section/' | uniq > override
# editor override # adjust priority e section
# dpkg-scanpackages pool override /usr/local/ \
   > dists/unstable/main/binary-i386/Packages
# cat > dists/unstable/main/Release << EOF
Archive: unstable
Version: 3.0
Component: main
Origin: Local
Label: Local
Architecture: i386
EOF
# echo "deb file:/usr/local unstable main" \
   >> /etc/apt/sources.list
```

In alternativa, un altro metodo per creare un archivio locale `deb`, molto alla bruta:

```
# aptitude install dpkg-dev
```

```
# mkdir /usr/local/debian
# mv /dove/è/pacchetto.deb /usr/local/debian
# dpkg-scanpackages /usr/local/debian /dev/null | \
  gzip - > /usr/local/debian/Packages.gz
#  echo "deb file:/usr/local/debian ./" >> /etc/apt/sources.list
```

Questi archivi possono essere raggiunti da remoto tramite HTTP od FTP, modificeo le voci in `/etc/apt/sources.list` di conseguenza.

3.4.12 Convertire od installare un pacchetto binario di altra distribuzione

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

3.4.13 Comando di installazione automatica

`auto-apt` è uno strumento di installazione dei pacchetti a richiesta

```
$ sudo auto-apt update
 ... aggiorna il database
$ auto-apt -x -y run
Entering auto-apt mode: /bin/bash
Exit the command to leave auto-apt mode.
$ less /usr/share/doc/med-bio/copyright # access non-existing file
 ...  Installa il pacchetto che fornisce questo file.
 ... Installa anche le dipendenze
```

3.4.14 Verificare i file dei pacchetti installati

`debsums` permette la verifica dei file dei pacchetti installati contro gli MD5 checksums. Alcuni pacchetti non hanno MD5 checksums. Una soluzione temporanea per gli amministratori di sistema:

```
# cat >>/etc/apt/apt.conf.d/90debsums
DPkg::Post-Install-Pkgs {"xargs /usr/bin/debsums -sg";};
^D
```

da Joerg Wendle <joergle@debian.org> (non testata).

3.5 Altre particolarità di Debian

3.5.1 Il comando `dpkg-divert`

Le **deviazioni** dei files rappresentano un modo di forzare `dpkg` a non installare un file nella sua posizione predefinita, ma in una posizione **deviata**. Le **deviazioni** possono essere utilizzate tramite gli script dei pacchetti Debian per muovere un file quando causa un conflitto. Gli amministratori di sistema possono usare una deviazione anche per scavalcare un file di configurazione di un pacchetto, oppure quando alcuni file (non marcati come conffiles) devono essere salvaguardati da `dpkg`, durante l'installazione di una nuova versione di un dato file.

```
# dpkg-divert [--add]  filename # aggiunge la "deviazione"
# dpkg-divert --remove filename # rimuove la "deviazione"
```

Di solito è una buona idea non usare `dpkg-divert` a meno che non sia strettamente necessario.

3.5.2 Il pacchetto `equivs`

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

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

3.5.3 Comandi alternativi

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

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

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

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

Per impostare il vostro ambiente X Window preferito, usate invece `update-alternatives` su `/usr/bin/x-session-manager` e `/usr/bin/x-window-manager`.

`/bin/sh` è un collegamento simbolico a `/bin/bash` o `/bin/dash`. E' più sicuro usare `/bin/bash` per mantenere la compatibilità con vecchi script in contaminati da bash, ma più corretto utilizzare `/bin/dash` per rinforzare la compatibilità POSIX. L'aggiornamento al kernel 2.4 tende ad impostare il collegamento su `/bin/dash`

3.5.4 Uso dei runlevel

Una volta installati, gran parte dei pacchetti Debian configura i propri servizi per girare nei runlevel da 2 a 5. Quindi, non esistono differenze fra i runlevel 2, 3, 4 e 5 su un sistema Debian non personalizzato; Debian lascia all'amministratore la possibilità di personalizzare i runlevel. Ciò differisce dal modo in cui i runlevel sono usati da altre distribuzioni popolarin GNU/Linux. Una modifica che potreste voler fare e disabilitare `xdm` o `gdm` nel runlevel 2 in maniera da non lanciare l'X display manager al termine della sequenza di boot; potete lanciarlo passando al runlevel 3.

3.5.5 Demoni di servizio disabilitati

Gli sviluppatori Debian hanno molto a cuore la sicurezza del sistema. Molti demoni di servizio sono installati con il minimo dei servizi abilitati.

Se avete dei dubbi (su Exim, DHCP, ...) controllate `ps aux` oppure il contenuto di `/etc/init.d/*` e di `/etc/inetd.conf`. Controllate anche `/etc/hosts.deny`. Anche il comando `pidof` è utile (vedere `pidof(8)`).

X11 non permette connessioni TCP/IP (remote) predefinite nelle versioni più recenti di Debian. Anche l'X forwarding in SSH è disabilitato.

Appendice A

Appendice

A.1 Autori

La guida Debian rapida è stata iniziata da Osamu Aoki <osamu\#at\#debian.org> come memo personale di installazione e successivamente è stata chiamata "Quick Reference . . . ". Buona parte del contenuto proviene dagli archivi della mailing list debian-user. Anche la "Debian – Installation Manual" e le "Debian – Release Notes" sono state utilizzate.

Seguendo il consiglio di Josip Rodin, che è molto attivo al Debian Documentation Project (http://www.debian.org/doc/ddp) (DDP) ed è anche l'attuale responsabile della "Debian FAQ", questo documento è stato rinominato "La guida Debian" ed è stato fuso con parecchi capitoli provenienti dalla "Debian FAQ" con un contenuto simil-bibliografico. Da qui "La guida Debian rapida" è stato formato come un estratto.

Questo documento stato scritto, tradotto ed espanso dai seguenti membri del gruppo QREF:

- Per il documento originale inglese "Quick Reference. . . "
  - Osamu Aoki <osamu\#at\#debian.org> (responsabile per tutto il progetto)
- Per la correzione delle bozze e la riscrittura inglese
  - David Sewell <dsewell\#at\#virginia.edu> (responsabile per la sezione en)
  - Thomas Hood <jdthood\#at\#yahoo.co.uk> (relativo alla rete)
  - Brian Nelson <nelson\#at\#bignachos.com>
  - Jan Michael C Alonzo <jmalonzo\#at\#spaceants.net>
  - Daniel Webb <webb\#at\#robust.colorado.edu>
  - Suggerimenti da tutti i traduttori
- Traduzione francese
  - Guillaume Erbs <gerbs\#at\#free.fr> (responsabile sez. fr)
  - Rénald Casagraude <rcasagraude\#at\#interfaces.fr>
  - Jean-Pierre Delange <adeimantos\#at\#free.fr>
  - Daniel Desages <daniel\#at\#desages.com>
- Traduzione italiana
  - Davide Di Lazzaro <mc0315\#at\#mclink.it> (responsabile sez. it)
- Traduzione Portoghese

  - Paulo Rogério Ormenese <pormenese\#at\#uol.com.br> (responsabile sez. pt-br)
  - Andre Luis Lopes <andrelop\#at\#ig.com.br>
  - Marcio Roberto Teixeira <marciotex\#at\#pop.com.br>
  - Rildo Taveira de Oliveira <to_rei\#at\#yahoo.com>
  - Raphael Bittencourt Simoes Costa <raphael-bsc\#at\#bol.com.br>
  - Gustavo Noronha Silva <kov\#at\#debian.org> (coordinator)
- Traduzione Spagnola
  - Walter Echarri <wecharri\#at\#infovia.com.ar> (responsabile sez. es)
  - José Carreiro <ffx\#at\#urbanet.ch>
- Traduzione Tedesca
  - Jens Seidel <tux-master\#at\#web.de> (responsabile sez. de)
  - Willi Dyck <wdyck\#at\#gmx.net>
  - Stefan Schröder <stefan\#at\#fkp.uni-hannover.de>
  - Agon S. Buchholz <asb\#at\#kefk.net>
- Traduzione Polacca—i seguenti membri del PDDP (http://debian.linux.org.pl):
  - Marcin Andruszkiewicz
  - Mariusz Centka <mariusz.centka\#at\#debian.linux.org.pl>
  - Bartosz Feński <fenio\#at\#debian.linux.org.pl> (resp. sez. pl)
  - Radosław Grzanka <radekg\#at\#debian.linux.org.pl>
  - Bartosz 'Xebord' Janowski
  - Jacek Lachowicz
  - Rafał Michaluk
  - Leonard Milcin, Jr.
  - Tomasz Z. Napierała <zen\#at\#debian.linux.org.pl>
  - Oskar Ostafin <cx\#at\#debian.linux.org.pl>
  - Tomasz Piękoś
  - Jacek Politowski
  - Mateusz Prichacz <mateusz\#at\#debian.linux.org.pl>
  - Marcin Rogowski
  - Paweł Różański
  - Mariusz Strzelecki
  - Krzysztof Ścierski
  - Przemysław Adam Śmiejek <tristan\#at\#debian.linux.org.pl>
  - Mateusz Tryka <uszek\#at\#debian.linux.org.pl>
  - Cezary Uchto
  - Krzysztof Witkowski <tjup\#at\#debian.linux.org.pl>
  - Bartosz Zapałowski <zapal\#at\#debian.linux.org.pl>
- Traduzione Cinese (semplificato)
  - Hao "Lyoo" LIU <iamlyoo\#at\#163.net>
  - Ming Hua <minghua\#at\#rice.edu>
  - Xiao Sheng Wen <atzlinux\#at\#163.com> (resp.: zh-cn)
  - Haifeng Chen <optical.dlz\#at\#gmail.com>
  - Xie Yanbo <xieyanbo\#at\#gmail.com>
  - easthero <easthero\#at\#gmail.com>
- Traduzione Cinese (tradizionale)

  - Asho Yeh <asho\#at\#debian.org.tw> (resp.: zh-tw)
  - Tang Wei Ching <wctang\#at\#csie.nctu.edu.tw> (ex resp.: zh-tw)
- Traduzione Giapponese
  - Shinichi Tsunoda <tsuno\#at\#ngy.1st.ne.jp> (resp.: ja)
  - Osamu Aoki <osamu\#at\#debian.org>

A.2 Garanzie

Dato che non sono un esperto, non pretendo di possedere la verità assoluta su Debian o Linux in generale. Le considerazioni sulla sicurezza che faccio sono solamente applicabili per un uso casalingo.

Questo documento non rimpiazza alcuna guida di riferimento.

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

A.3 Feedback

Commenti ed aggiunte a questo documento sono sempre i benvenuti. Mandate una mail al Sistema Debian di tracciamento dei bachi (`http://bugs.debian.org/`) relativa al pacchetto `debian-reference`, o ai pacchetti delle rispettive trauzioni. L'uso di `reportbug` semplifica l'invio di un rapporto di un baco. Potete sempre mandare una email a Osamu Aoki (`http://people.debian.org/~osamu/`) at <osamu\#at\#debian.org> in inglese, oppure ai traduttori delle rispettive lingue.