

La gestione dei pacchetti in Ubuntu Server

Installare e disinstallare i pacchetti

Autore: Alessandro Di Nicola

Tratto dal libro: [Linux Ubuntu per server e reti](#)

Nello scorso articolo abbiamo visto come gestire gli utenti e i loro permessi. Un'altra parte importante nella gestione di un server è l'amministrazione dei pacchetti e delle repository, in questo articolo vedremo come fare.

La gestione dei pacchetti

In Ubuntu Server i programmi vengono forniti all'interno di appositi pacchetti: ogni pacchetto è costituito dall'insieme dei file che verranno effettivamente installati sul sistema e da alcuni file di controllo che contengono le istruzioni per installare e disinstallare in modo pulito il pacchetto stesso. I pacchetti si relazionano gli uni con gli altri. Per esempio, un web browser necessita di alcune librerie per funzionare (in gergo si dice che il browser "dipende" da queste librerie): il web browser sarà presente in un pacchetto e le librerie richieste saranno contenute in altri pacchetti ancora. In questo caso, programma e librerie devono essere perfettamente compatibili tra loro: per ottenere questo, i relativi pacchetti vengono prelevati da un repository comune, cioè da un archivio contenente un ampio numero di pacchetti dalla compatibilità reciproca certa. L'insieme dei repository attivi in una macchina, dunque, forma un ecosistema armonioso in cui ogni singolo elemento si integra con le parti restanti: la versione del kernel è adatta alle librerie di sistema installate, le singole applicazioni alle diverse librerie presenti e così via.

Installare e cercare pacchetti

Il comando principale per gestire i pacchetti nel sistema è `apt-get`. Per installare un nuovo pacchetto, quindi, la sintassi da usare è la seguente:

```
sudo apt-get install <pacchetto>
```

```
ale@pitagora:~$ sudo apt-get install apache2
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following extra packages will be installed:
  apache2-mpm-worker apache2-utils apache2.2-common libapr1 libaprutil1
Suggested packages:
  apache2-doc
The following NEW packages will be installed:
  apache2 apache2-mpm-worker apache2-utils apache2.2-common libapr1
  libaprutil1
0 upgraded, 6 newly installed, 0 to remove and 0 not upgraded.
Need to get 1355kB of archives.
After this operation, 4878kB of additional disk space will be used.
Do you want to continue [Y/n]? █
```

Figura 1: con il comando `sudo apt-get install` si installano nuovi pacchetti dai repository

Per installare il pacchetto chiamato **emacs**, per esempio, si userà quindi il comando `sudo apt-get install emacs`.

Dopo aver premuto **Invio** per conferma, il pacchetto verrà scaricato da Internet e installato nel sistema, insieme a tutti i pacchetti da cui dipende (le “dipendenze”) per il proprio corretto funzionamento.

Per ricercare un pacchetto tra tutti quelli disponibili si usa invece il comando `apt-cache`, così:

```
apt-cache search <criterio>
```

Al posto di `<criterio>` si inserisce il testo che va ricercato tra i nomi dei pacchetti e tra le descrizioni di questi.

Per esempio, per cercare tutte le occorrenze di “**web server**” si lancia:

```
apt-cache search web server
```

L’output scorrerà nella console e l’elenco dei vari pacchetti che soddisfano il criterio inserito sarà visualizzato senza seguire alcun ordine. Per avere i risultati della ricerca in ordine alfabetico, invece, si digiti il comando seguente:

```
apt-cache search web server | sort | less
```

Nota

Il comando appena eseguito è una pipe: il risultato di `apt-cache search` diventa l’input di `sort` (che ordina i dati), quindi l’output di `sort` diventa l’input di `less` (che visualizza i dati in schermate separate).

In questo caso, inoltre, i risultati non scorreranno riga per riga in console, ma verranno mostrati una schermata per volta: per passare alla schermata successiva basterà premere il tasto **Spazio**.

Per ottenere un elenco dei pacchetti installati nel sistema si usa il comando seguente:

```
dpkg -l
```

Aggiungendo come argomento dell’opzione `-l` una qualsiasi stringa seguita dal carattere `*`, verranno mostrati solo i pacchetti installati i cui nomi contengono la stringa inserita. Quindi, per elencare tutti i pacchetti che hanno nel nome “`libc`” si dovrà scrivere questo in console:

```
dpkg -l libc*
```

Nota

Il carattere `` della riga `dpkg -l libc*` è un carattere speciale (metacarattere) e significa “qualsiasi successione di caratteri, anche vuota”. L’argomento `libc*`, quindi, viene tradotto dal comando `dpkg` in “i caratteri `l`, `i`, `b` e `c` seguiti da una qualsiasi successione di caratteri”. Una sequenza come `libc*` è chiamata **glob pattern**.*

Informazioni complete

Una volta che si è individuato un pacchetto con `apt-cache search`, si ha spesso la necessità di ottenere informazioni dettagliate su di esso. Per fare questo si usa il comando interno `show` di `apt-cache`, così:

```
apt-cache show <pacchetto>
```

Nelle righe di output visualizzate in console da `apt-cache show` sono presenti numerosi campi; ecco i principali:

- **Version:** indica la release del pacchetto presente nel repository;
- **Depends:** contiene l'elenco di tutte le dipendenze del pacchetto;
- **Suggests:** sono i pacchetti suggeriti: se le dipendenze consentono di far funzionare il programma, i pacchetti suggeriti permettono al programma di funzionare al meglio;
- **Conflicts:** i pacchetti in conflitto con il pacchetto selezionato. Non è possibile installare contemporaneamente questo e quelli;
- **Description:** una descrizione accurata del software presente nel pacchetto.

Se nei repository sono presenti più versioni di uno stesso pacchetto, il comando `apt-cache show` mostrerà informazioni su tutte le release disponibili, in sequenza.

Pacchetti sempre aggiornati

In una macchina server è indispensabile avere sempre le ultime versioni dei pacchetti rilasciate dalla distribuzione. In questo modo si starà ragionevolmente al riparo da bug insidiosi che possano compromettere la sicurezza del server.

L'operazione di aggiornamento consta di due passaggi. Prima si scarica da Internet la lista più recente dei pacchetti disponibili:

```
sudo apt-get update
```

Fatto questo, per aggiornare tutti i pacchetti installati all'ultima release si lancia il comando seguente:

```
sudo apt-get upgrade
```

```
ale@pitagora:~$ sudo apt-get update
Hit http://it.archive.ubuntu.com hardy Release.gpg
Hit http://security.ubuntu.com hardy-security Release.gpg
Hit http://archive.canonical.com hardy Release.gpg
Hit http://it.archive.ubuntu.com hardy-updates Release.gpg
Hit http://security.ubuntu.com hardy-security Release
Hit http://archive.canonical.com hardy Release
Hit http://security.ubuntu.com hardy-security/main Packages
Hit http://it.archive.ubuntu.com hardy Release
Hit http://archive.canonical.com hardy/partner Packages
Hit http://security.ubuntu.com hardy-security/restricted Packages
Hit http://security.ubuntu.com hardy-security/main Sources
Hit http://security.ubuntu.com hardy-security/restricted Sources
Hit http://it.archive.ubuntu.com hardy-updates Release
Hit http://security.ubuntu.com hardy-security/universe Packages
Hit http://security.ubuntu.com hardy-security/universe Sources
Hit http://security.ubuntu.com hardy-security/multiverse Packages
Hit http://security.ubuntu.com hardy-security/multiverse Sources
Hit http://it.archive.ubuntu.com hardy/main Packages
Hit http://it.archive.ubuntu.com hardy/restricted Packages
Hit http://it.archive.ubuntu.com hardy/main Sources
Hit http://it.archive.ubuntu.com hardy/restricted Sources
Hit http://it.archive.ubuntu.com hardy/universe Packages
Hit http://it.archive.ubuntu.com hardy/universe Sources
```

Figura 2: eseguendo `sudo apt-get update` si scarica una lista aggiornata dei pacchetti disponibili

La politica di aggiornamento di `apt-get upgrade` è piuttosto stringente: con questo comando nessun pacchetto sarà disinstallato o aggiunto al sistema anche se ciò viene richiesto per portare a termine l'aggiornamento di uno o più pacchetti. Questo comando, dunque, consente di mantenere l'assoluto controllo sul parco software presente in una macchina, ma non assicura che vengano aggiornati tutti i pacchetti installati.

A volte, però, il completo aggiornamento di un sistema richiede l'installazione di pacchetti aggiuntivi o l'eliminazione di altri: in questi casi si deve utilizzare il comando `sudo apt-get dist-upgrade` che, in caso di conflitti tra i pacchetti, aggiornerà quelli considerati più importanti a scapito degli altri.

Eliminare i pacchetti

Per cancellare un pacchetto dal sistema insieme a tutti i file che lo compongono si usa `apt-get` seguito dal comando interno `remove`, in questo modo:

```
sudo apt-get remove <pacchetto>
```

L'installazione di un pacchetto comporta spesso la creazione di alcuni file di configurazione sull'hard disk: questi non vengono eliminati dal comando `apt-get remove`. Per cancellare sia i file effettivi di un pacchetto sia i file di configurazione, dunque, si deve utilizzare `apt-get purge`:

```
sudo apt-get purge <pacchetto>
```

Quando si installa un pacchetto, il file di questo viene memorizzato nella directory `/var/cache/apt/archives`. Ciò può essere utile per reinstallare rapidamente dei pacchetti che si sono installati in passato: un pacchetto presente in questa directory, infatti, non verrà scaricato nuovamente dalla rete. Dopo aver installato molti nuovi pacchetti e aggiornato più volte la macchina, però, è inevitabile che la directory `/var/cache/apt/archives` risulti piena di file `deb` e che, perciò, finisca per occupare centinaia di MB di spazio su disco.

Note

*I file dei pacchetti di Ubuntu Server hanno estensione **.deb**, dato che il sistema di gestione dei pacchetti di Ubuntu è lo stesso presente nella distribuzione **Debian**. Visto il suffisso finale dei file, tali pacchetti vengono chiamati "pacchetti deb".*

```
ale@pitagora:~$ ls /var/cache/apt/archives/  
compiz-core_1%3a0.7.4-0ubuntu7_i386.deb  
compiz-gnome_1%3a0.7.4-0ubuntu7_i386.deb  
compiz-plugins_1%3a0.7.4-0ubuntu7_i386.deb  
compiz_1%3a0.7.4-0ubuntu7_all.deb  
evolution-common_2.22.2-0ubuntu2_all.deb  
evolution-plugins_2.22.2-0ubuntu2_i386.deb  
evolution_2.22.2-0ubuntu2_i386.deb  
googleearth_4.3.7204.836-0medibuntu1_all.deb  
gtwitter_1.0-beta-6ubuntu1_all.deb  
libdecoration0_1%3a0.7.4-0ubuntu7_i386.deb  
libgl2.0-0_2.16.3-1ubuntu3_i386.deb  
libgl2.0-data_2.16.3-1ubuntu3_all.deb  
libldap-2.4-2_2.4.9-0ubuntu0.8.04_i386.deb  
libpoppler-glib2_0.6.4-1ubuntu2_i386.deb  
libpoppler-qt2_0.6.4-1ubuntu2_i386.deb  
libpoppler2_0.6.4-1ubuntu2_i386.deb  
libsoup2.2-8_2.2.105-4_i386.deb  
libssl0.9.8_0.9.8g-4ubuntu3.3_i386.deb  
linux-headers-2.6.24-19-generic_2.6.24-19.34_i386.deb  
linux-headers-2.6.24-19_2.6.24-19.34_all.deb  
linux-image-2.6.24-19-generic_2.6.24-19.34_i386.deb  
linux-source-2.6.24_2.6.24-19.34_all.deb  
linux-ubuntu-modules-2.6.24-19-generic_2.6.24-19.28_i386.deb
```

Figura 3: Se `/var/cache/apt/archives` è piena, con `sudo apt-get clean` si cancellano tutti i file dalla directory

È possibile "sfolire" `/var/cache/apt/archives` in due modi: cancellando tutti i file presenti nella cartella oppure eliminando solo quei file dei pacchetti che non sono più scaricabili dai repository su

Internet (e dei quali, dunque, non risulta più di alcuna utilità avere una copia sull'hard disk). Per eliminare tutti i file deb da `/var/cache/apt/archives` va usato il comando seguente:

```
sudo apt-get clean
```

Con `sudo apt-get autoclean`, invece, verranno cancellati solo i file di cui non è più possibile effettuare il download.

Gestire i repository

Nella configurazione di default di Ubuntu Server sono attivi tutti i repository necessari per mantenere costantemente aggiornato il proprio server. È possibile, comunque, gestire direttamente i repository abilitati nel sistema intervenendo sul file `/etc/apt/sources.list`, un semplice file di testo che va aperto con un editor.

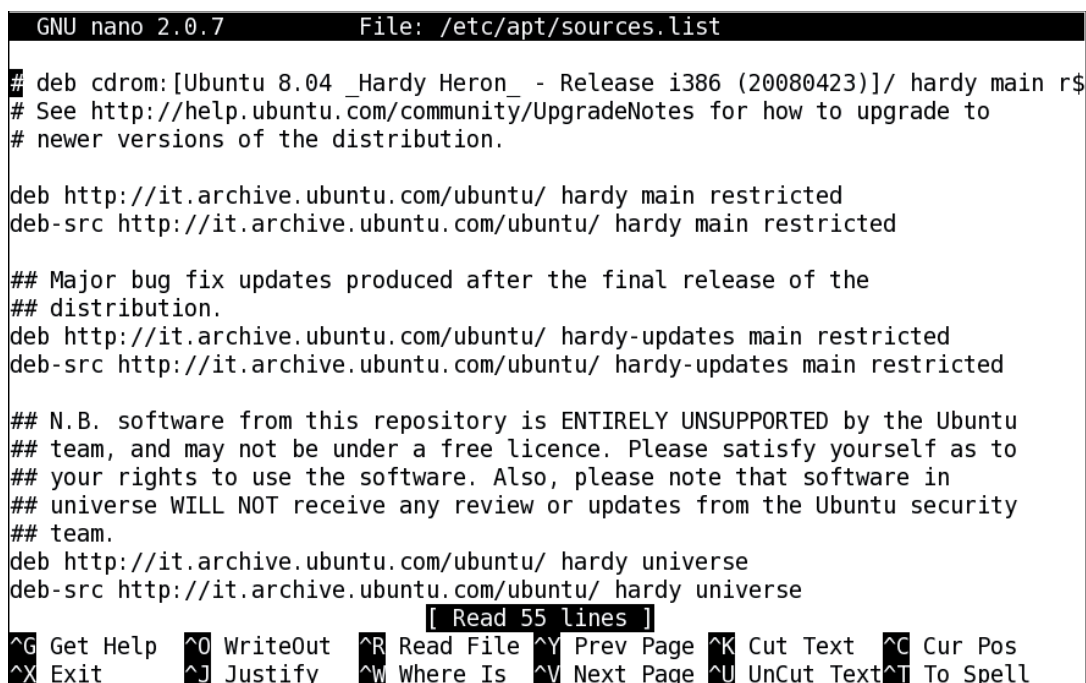
Molto del lavoro di amministrazione di un server viene svolto all'interno di un editor di testo: i file di configurazione del sistema o quelli dei programmi, infatti, non sono altro che file testuali modificabili dagli utenti provvisti dei permessi necessari.

La presentazione del file `/etc/apt/sources.list`, quindi, offrirà l'occasione giusta per conoscere da vicino uno degli editor installati per default su Ubuntu Server: nano.

L'editor nano

Per aprire il file `/etc/apt/sources.list` con l'editor **nano** è necessario far uso di `sudo`, dato che un utente normale non avrebbe i permessi per modificare il file. Si lanci quindi in console il comando seguente:

```
sudo nano /etc/apt/sources.list
```



```
GNU nano 2.0.7 File: /etc/apt/sources.list
# deb cdrom:[Ubuntu 8.04 _Hardy Heron_ - Release i386 (20080423)]/ hardy main rs
# See http://help.ubuntu.com/community/UpgradeNotes for how to upgrade to
# newer versions of the distribution.
deb http://it.archive.ubuntu.com/ubuntu/ hardy main restricted
deb-src http://it.archive.ubuntu.com/ubuntu/ hardy main restricted
## Major bug fix updates produced after the final release of the
## distribution.
deb http://it.archive.ubuntu.com/ubuntu/ hardy-updates main restricted
deb-src http://it.archive.ubuntu.com/ubuntu/ hardy-updates main restricted
## N.B. software from this repository is ENTIRELY UNSUPPORTED by the Ubuntu
## team, and may not be under a free licence. Please satisfy yourself as to
## your rights to use the software. Also, please note that software in
## universe WILL NOT receive any review or updates from the Ubuntu security
## team.
deb http://it.archive.ubuntu.com/ubuntu/ hardy universe
deb-src http://it.archive.ubuntu.com/ubuntu/ hardy universe
[ Read 55 lines ]
^G Get Help ^O WriteOut ^R Read File ^Y Prev Page ^K Cut Text ^C Cur Pos
^X Exit ^J Justify ^W Where Is ^V Next Page ^U UnCut Text ^T To Spell
```

Figura 4: L'editor nano con aperto il file `/etc/apt/sources.list`

Comparirà una schermata con in alto una riga informativa contenente il nome del file aperto, in basso un riepilogo dei comandi disponibili nell'editor e, nella sezione principale al centro, il contenuto del file `/etc/apt/sources.list`. Il risultato su schermo sarà simile all'immagine visibile nella figura 4.

L'editor **nano** è molto semplice da usare: con i **tasti freccia** ci si sposta all'interno del file aperto, con la combinazione di tasti **Ctrl+O** si salva il file e con **Ctrl+X** si esce dall'editor. Premendo **Ctrl+G**, quindi, viene fornito un prospetto dei principali comandi disponibili in **nano**, con le relative combinazioni di tasti.

Il file `sources.list` per i repository

Prendendo come riferimento il file `/etc/apt/sources.list` di default su Ubuntu Server, le prime righe sono righe di commento: iniziano con il carattere `#` e consentono di aggiungere informazioni sugli elementi di configurazione presenti nel file.

Nota

Può stupire che all'interno di un semplice file di configurazione siano presenti delle righe di commento, come nei sorgenti dei programmi. In realtà, data l'importanza dei file di configurazione nell'amministrazione di una macchina, capiterà spesso di aprirne uno a distanza anche di anni dalle ultime modifiche apportate: delle chiare righe di commento saranno allora fondamentali per potersi orientare.

Le righe effettive di configurazione forniscono i dati sui repository da attivare e utilizzano il seguente formato:

```
deb uri distribuzione componenti
```

Ogni linea rappresenta uno o più repository. Il primo elemento ne indica il tipo: **deb** per i repository di pacchetti con i file eseguibili compilati, **deb-src** per quelli contenenti i sorgenti dei pacchetti. La voce **uri** contiene il percorso iniziale dei repository. Si tratta solitamente dell'indirizzo Internet da cui prelevare i pacchetti, privo della parte che specifica la versione della distribuzione cui si è interessati. L'elemento distribuzione, quindi, indica la release della distro installata sulla macchina: `hardy`, `gutsy`, `dapper` e tutti gli altri nomi attribuiti alle varie versioni di Ubuntu.

Nota

A ogni nuova release di Ubuntu viene affibbiato un nomignolo composto da due parole: un aggettivo e un nome di animale. In lingua inglese entrambe le parole cominciano con le stesse iniziali. Un elenco dei nomi delle diverse versioni si trova alla pagina <https://wiki.ubuntu.com/Releases>.

Per ogni release di Ubuntu, poi, sono disponibili sui server dei repository alcune cartelle che forniscono elementi distribuzione aggiuntivi.

Ad esempio per la versione 8.04 LTS di Ubuntu ha `hardy-updates`, contenente aggiornamenti che risolvono bug importanti nei pacchetti, e `hardy-backports`, che presenta versioni più recenti dei pacchetti forniti con il rilascio di quella distribuzione.

L'ultimo parametro, `componenti`, è formato dal nome del componente finale di uno o più repository che si desidera abilitare: per indicare più componenti è necessario che questi abbiano **uri** e **distribuzione** in comune.

Nella Tabella 1, visibile nella pagina seguente, vengono mostrati i principali componenti disponibili.

Tabella 1: elenco dei componenti

Componente	Significato
main	Pacchetti open source direttamente supportati da Ubuntu. Gli aggiornamenti di sicurezza avvengono in tempi molto rapidi.
universe	Pacchetti open source che, a differenza del componente main, sono gestiti dalla comunità di utenti. Gli aggiornamenti di sicurezza sono forniti dalla comunità.
multiverse	Pacchetti non open source. Questo componente e i relativi aggiornamenti di sicurezza sono gestiti dalla comunità di utenti.
restricted	Pacchetti non disponibili sotto licenza libera ma direttamente supportati, per quanto possibile. Contiene driver per hardware non altrimenti supportato su Linux.
partner	Pacchetti commerciali.

L'insieme di tutti questi elementi forma quindi una riga in `/etc/apt/sources.list` come la seguente:

```
deb http://it.archive.ubuntu.com/ubuntu/ hardy main restricted
```

Un repository a parte è <http://security.ubuntu.com/ubuntu>, che contiene i soli aggiornamenti di sicurezza per la distribuzione. Ecco una riga che lo richiama:

```
deb http://security.ubuntu.com/ubuntu hardy-security main restricted
```

La presenza di più repository per gli upgrade, ognuno dedicato a un diverso tipo di aggiornamento per i pacchetti, consente di scegliere in modo granulare cosa aggiornare o meno su di una determina macchina: una possibilità fondamentale in un server.

Modifiche ai repository

Ora che la struttura di `sources.list` è nota, si possono apportare delle modifiche alla configurazione di default del file in Ubuntu Server.

Per esempio, se non si è interessati alla compilazione dei sorgenti si possono disattivare tutti i repository di tipo `deb-src`: in questo modo la procedura di aggiornamento con `sudo apt-get update` sarà più rapida. Per ottenere questo basta inserire il carattere `#` all'inizio di ogni riga `deb-src`. Quindi, la linea seguente:

```
deb-src http://it.archive.ubuntu.com/ubuntu/ hardy main restricted
```

diventa

```
#deb-src http://it.archive.ubuntu.com/ubuntu/ hardy main restricted
```

In un server può essere utile avere disponibile solo il software direttamente supportato da Ubuntu: in tal modo si sarà certi che, qualunque applicazione si scelga di installare sulla propria macchina, in caso di bug accertati questi verranno corretti nel più breve tempo possibile. Per ottenere una configurazione di `sources.list` di tal fatta, quindi, è sufficiente commentare tutte le righe nel file che facciano riferimento ai componenti `universe` e `multiverse`.

Una volta che sono state apportate delle modifiche al file `sources.list`, bisogna poi aggiornare la lista dei pacchetti nei repository con `sudo apt-get update`.

Se si desidera aggiungere dei repository ma non si vuole "sporcare" il file `sources.list` principale, infine,

è possibile creare nella directory `/etc/apt/sources.list.d` dei file separati contenenti le righe necessarie per attivare i repository aggiuntivi.

Attenzione

Si controlli scrupolosamente la provenienza e il contenuto di questi repository per non compromettere la sicurezza della propria macchina.

Se il repository non risponde

Durante la procedura di installazione del sistema operativo vengono inseriti in `sources.list` dei mirror dei repository di Ubuntu geograficamente vicini alla macchina: per l'Italia si tratta del mirror con uri: <http://it.archive.ubuntu.com/ubuntu/>

Nota

Un mirror è la copia esatta del contenuto di un sito. In questo caso, si tratta di server http e ftp sparsi in tutto il mondo che replicano il contenuto degli archivi di repository Ubuntu.

Una scelta del genere, solitamente, permette di scaricare liste e pacchetti dai repository nei tempi più rapidi possibili; qualche volta, però, gli indirizzi predefiniti possono risultare irraggiungibili o semplicemente intasati.

In questi casi si può aprire la pagina web che elenca tutti i mirror disponibili. Per fare questo si può usare il browser testuale **w3m** installato per default su Ubuntu Server:

```
w3m https://wiki.ubuntu.com/Mirrors?action=show
```

Nell'elenco che appare, quindi, si individua un secondo mirror nelle vicinanze. Poi si devono cambiare in `sources.list` tutte le occorrenze di <http://it.archive.ubuntu.com/ubuntu/> inserendo al posto di queste l'uri del mirror alternativo. Per esempio, una riga come la seguente:

```
deb http://it.archive.ubuntu.com/ubuntu/ hardy universe
```

può diventare così:

```
deb ftp://ftp.fu-berlin.de/linux/ubuntu/ hardy universe
```

```
2. CD Images
   1. CD Image Mirrors
3. DVD Images
   1. DVD Image Mirrors

Sub-pages

Mirror-Mirrors

The following package archive mirrors are available, for you to mirror from, if
you would rather mirror from a downstream mirror than from the master servers
at Ubuntu:

(See also the mirror list at launchpad: [WWW] https://launchpad.net/ubuntu/
+archivemirrors )

Australia

• [WWW] http://ftp.iinet.net.au/pub/ubuntu/
• [WWW] http://mirror.optus.net/ubuntu/

<< ↑ ↓ Viewing[SSL] <Mirrors - Ubuntu Wiki>
```

Figura 5: Alla pagina <https://wiki.ubuntu.com/Mirrors?action=show> è disponibile un elenco dei mirror Ubuntu

Il gestore semigrafico dei pacchetti

È disponibile per la console una pratica interfaccia semigrafica per il gestore di pacchetti, chiamata **aptitude**.

La si avvia con il comando `sudo aptitude`. L'aspetto del programma è mostrato nell'immagine visibile in *figura 6*.

I pacchetti sono organizzati in sezioni e sottosezioni: per muoversi tra queste si usano i tasti **freccia su** e **freccia giù**, mentre per aprire la sezione o la sottosezione corrente si preme Invio. Individuato un pacchetto che si vuole installare si preme **+** per selezionarlo; per disinstallare un pacchetto si preme **-**. Con il tasto **u** si scaricano dai server remoti le liste aggiornate dei pacchetti e con **U**, invece, si selezionano tutti i pacchetti di cui sono disponibili aggiornamenti.

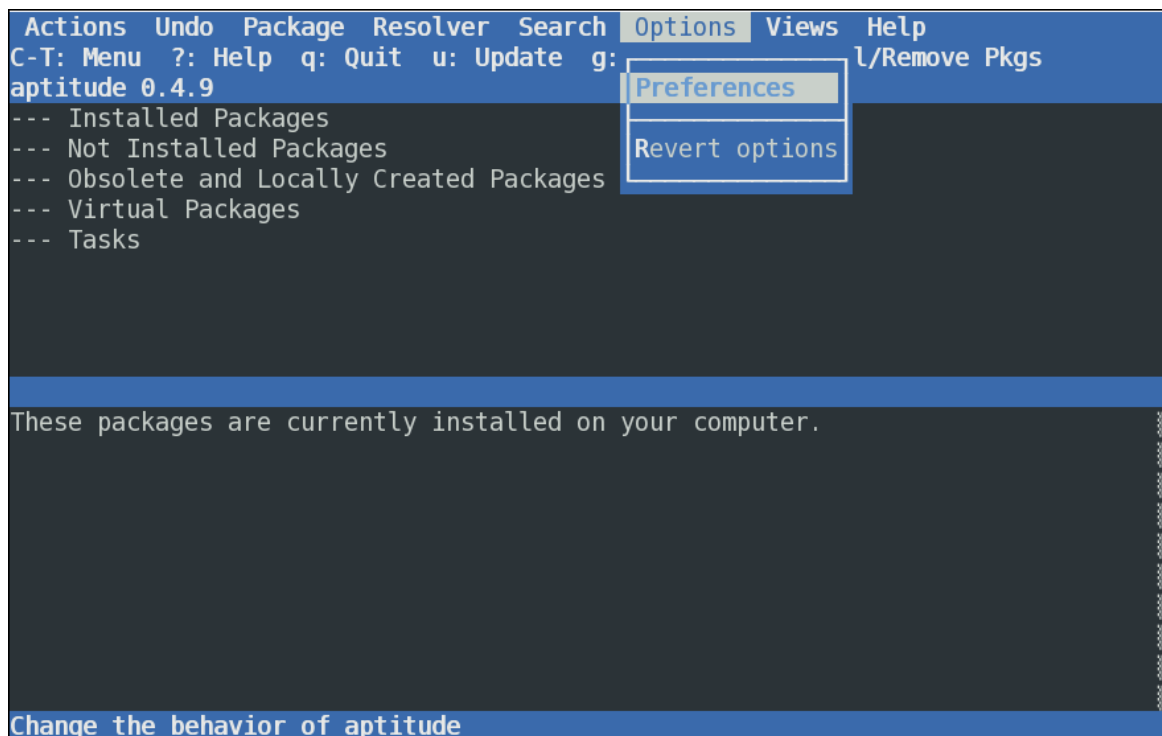


Figura 6: *aptitude* è un'interfaccia semigrafica per il gestore di pacchetti

Quindi, per procedere nelle operazioni di installazione, disinstallazione o aggiornamento si preme **g**. **Aptitude** dispone poi di una serie di menu che consentono l'accesso ad alcune funzionalità avanzate nell'amministrazione dei pacchetti: questi menu sono raggiungibili con la combinazione di tasti **Ctrl+T**. Come esempio, ecco la procedura da seguire per installare l'editor **emacs** tramite **aptitude**: con i tasti freccia si entra nella sezione **Not Installed Packages**, quindi si preme **Invio**, si apre la sottosezione **editors**, poi si entra in **main**. A questo punto si seleziona la riga con la scritta **emacs**, si preme **+** e poi **g**. Il pacchetto principale di **emacs** con tutte le dipendenze necessarie verrà installato nel sistema.

Come installare singoli pacchetti

Fino a ora si è visto come installare pacchetti contenuti nei repository. Per installare singoli pacchetti, scaricati da Internet o creati dalla compilazione di sorgenti, si usa il comando `dpkg` seguito dal file `deb` del pacchetto indicato per esteso e, se necessario, con il percorso completo per raggiungerlo.

Ecco un esempio:

```
sudo dpkg -i /tmp/unrar_1%3a3.7.8-1_i386.deb
```

A differenza di `apt-get` e `aptitude`, `dpkg` agisce a un livello più basso e non gestisce la risoluzione delle dipendenze: se il pacchetto indicato richiede delle dipendenze, quindi, sarà necessario procedere manualmente alla loro installazione.

Per installare pacchetti svincolati dai repository e, al tempo stesso, risolvere le dipendenze si può usare il programma `gdebi`. Dopo averlo aggiunto al sistema con `sudo apt-get install gdebi-core`, per installare un pacchetto si utilizzerà il comando nel modo seguente:

```
sudo gdebi /tmp/unrar_1%3a3.7.8-1_i386.deb
```

Alcuni pacchetti d'uso comune

Al di là del particolare utilizzo che si farà della macchina, ci sono dei pacchetti non installati per default su Ubuntu Server che è molto comodo avere a propria disposizione.

L'utilizzo di un file manager può rendere molto più rapide alcune operazioni di manutenzione, anche in un server: per la console è disponibile l'ottimo `Midnight Commander`, contenuto nel pacchetto `mc`.

Una volta installato il pacchetto, il file manager si lancia con il comando `mc`.

Questo gestore di file presenta la classica struttura a due pannelli, uno per la directory d'origine e l'altro per la directory di destinazione. Le operazioni sui file si compiono premendo i tasti funzione: **F5** per copiare, **F6** per spostare, **F8** per cancellare e così via.

Un altro pacchetto utile è `language-pack-it`, che consente di avere la traduzione completa in lingua italiana dei messaggi e dell'interfaccia di molti dei principali programmi disponibili.

Inoltre, installando il pacchetto `manpages-it` anche le pagine di manuale dei comandi saranno in italiano.

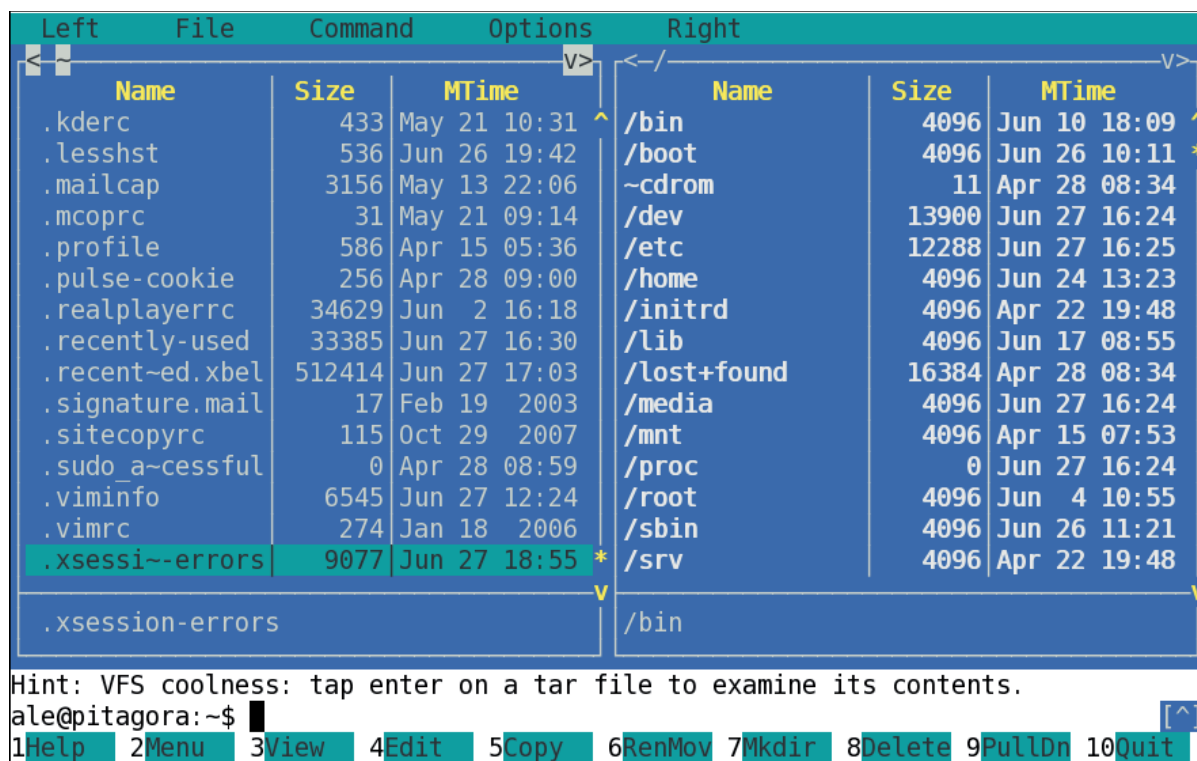


Figura 7: l'interfaccia del file manager `mc`

Conclusioni

In questo articolo su Linux Ubuntu Server abbiamo visto come si amministra l'installazione di pacchetti singoli o tramite repository.



Resta sempre aggiornato sulle novità del sito

Per mantenerti sempre aggiornato su nuovi contenuti interessanti, Come fare a... vi offre la possibilità di abbonarvi gratuitamente alla **Newsletter Come fare a...** all'indirizzo: <http://www.comefarea.it/newsletter/> o, se usate Windows Live Messenger, di abbonarvi ai nostri **Windows Live Alerts** all'indirizzo <http://www.comefarea.it/abbonamenti.php>. Per gli utenti di Mac OS X Leopard è anche disponibile gratuitamente un Widget che vi terrà sempre informati sulle ultime novità

Il libro



Linux Ubuntu per server e reti

Quando si pensa al sistema operativo Linux in ambito server si pensa solitamente ad austeri centri di calcolo, ad ambienti di ricerca all'avanguardia e a enormi reti di computer. Un server Linux, però, può essere anche un comune PC e una rete di computer può essere costituita, perché no, da una manciata di macchine. Una rete di tal fatta può venire sfruttata per condividere una medesima stampante fra più PC o per creare un file server grazie al quale i membri di un team di lavoro possono scambiarsi documenti con estrema facilità. Singole macchine, poi, possono essere adibite al ruolo di web server e mail server oppure, ancora, possono venire impiegate per allestire dei media center. Questo libro spiega come costruire dalle fondamenta reti e server basati su Linux, utilizzando come

guida la distribuzione Ubuntu Server: la scelta di un'unica distribuzione consente di fornire informazioni mirate su strumenti specifici e di guidare passo per passo il lettore anche nelle procedure più articolate, per quanto molte indicazioni di carattere metodologico fornite nel libro abbiano un valore generale.

Come acquistare il libro

Se desideri acquistare questo libro, puoi farlo direttamente online sul sito di FAG al seguente indirizzo: <http://www.fag.it/scheda.aspx?ID=28755>

La recensione

Per conoscere e approfondire gli argomenti trattati in questo articolo è disponibile la recensione del libro da cui è tratto. Visita la pagina "Linux Ubuntu per server e reti", disponibile all'indirizzo: <http://www.comefarea.it/recensioni/linuxubuntuserverereti/>.

L'autore

Alessandro Di Nicola collabora con riviste di informatica come Linux Pro e Linux Magazine, per la quale redige mese per mese l'articolo di copertina, ed è dottore in Lettere moderne.