

20 Devo disinstallare un programma

Abbiamo imparato molte cose su come gestire i programmi Linux.

Ricapitolando, abbiamo detto che di norma, su Red Hat e distribuzioni simili, i programmi vengono distribuiti in pacchetti. Ogni pacchetto ha un nome che è quello del programma più un numero di versione più o meno complesso: ad esempio `mplayer-common-0.90-2.rpm` è la versione 0.90 (seconda compilazione) del pacchetto `mplayer-common`. Dal nome si intuisce che questa è la parte di base di un pacchetto più complesso di nome `mplayer`.

Una gestione di questo tipo non è stata studiata per complicare la vita all'utente, che ovviamente preferirebbe avere dei pacchetti autoinstallanti con un semplice click del mouse, tipo Windows, ma serve per tenere tutto in ordine.

Mentre in Windows ogni programma si può installare senza rivelarsi al sistema, finché noi installiamo rpm, i programmi si dichiarano al sistema operativo (o meglio alla componente dedicata alla gestione del pacchetto), dicendo nome, versione, da che altri pacchetti eventualmente dipendono, e cosa offrono (per esempio una libreria chiamata `libc`). E' dunque possibile tenere un database di tutte le applicazioni presenti sul computer, tenendo conto di dove sono esattamente installate, e di quali altre applicazioni si basano su di esse. Di più: è immaginabile la possibilità di creare dei database giganteschi in Internet che tengano conto di tutti i possibili pacchetti creati per ogni distribuzione RedHat, e che permettano l'installazione via rete con risoluzione di dipendenze in maniera semi automatica.

Red Carpet, che abbiamo usato qualche volta, si basa su questo concetto, ma in piccolo: gestisce infatti tutti i pacchetti prodotti da RedHat, ed in più (a scelta) quelli prodotti dalla Ximian per il suo desktop Gnome.

Il sito `rpmfind.net`, invece, ha un database molto più grande, ma si limita a trovare i singoli pacchetti, ad elencarne i file e le dipendenze, e a farli scaricare uno ad uno, lasciando all'utente il compito di riunire tutto il software che gli è (anche indirettamente) necessario, e di installarlo.



Rimuovere un pacchetto è dunque uno dei punti critici del sistema: per come è costruito per minimizzare la dimensione di ogni singolo pacchetto, se nel pacchetto che per qualche motivo vogliamo rimuovere è contenuto anche un solo file necessario ad un altro pacchetto, saremo costretti ad eliminare entrambi, o nessuno.

E' duro abituarsi le prime volte, essendo abituati al mondo Microsoft in cui tutto è permesso (anche distruggere il sistema per delle sciocchezze) dandoci una sorta d'idea di onnipotenza.

Il modo più sicuro di rimuovere un pacchetto è quello di usare la gestione pacchetti di Red Hat o il Red Carpet, che controllano automaticamente le dipendenze, e ci avvertono di cosa dovremmo togliere se vogliamo proseguire nel nostro intento (togliendo tutto con un semplice click).

D'altra parte questo meccanismo può essere molto comodo in casi estremi: supponendo per esempio di volere disinstallare tutti i programmi grafici perché ci siamo infatuati della modalità testuale, non dovremo togliere i 500 pacchetti circa relativi a programmi con interfacce grafiche: molto più semplicemente basterà eliminare Xfree86, perché tutto il resto segua in maniera automatica!

Qualora non si riesca a trovare il pacchetto nella lunga lista che ci viene proposta (anche se Red Carpet ci offre l'opportunità per fare la ricerca sulla lista, eliminando anche questo problema), basterà aprire una shell, diventare root (solito comando 'su') e digitare:

```
rpm -e nomePacchetto
```

in cui il nome del pacchetto non include la versione, e nemmeno l'estensione .rpm. Se per esempio volessimo eliminare il pacchetto contenente il browser testuale lynx, sarà sufficiente digitare 'rpm -e lynx'. Se il pacchetto dovesse essere indispensabile per altri programmi, allora il comando terminerebbe con un errore di dipendenze fallite, dando la lista dei programmi che hanno bisogno di ciò che noi vogliamo eliminare, che andrebbero eventualmente tolti uno per uno se davvero vogliamo completare l'operazione.

Supponiamo ora di voler conoscere a quale pacchetto appartiene uno specifico file, non necessariamente per cancellarlo: la tecnica di inserire tutte le informazioni sui pacchetti in un database permette di gestire in modo assolutamente efficiente una qualsiasi interrogazione (o query, per dirla in gergo da database) su di esso.

Basterà pertanto digitare

```
rpm -qf nomeFile
```

per vedersi rispondere in un tempo di solito molto breve (provate ad esempio a digitare rpm -qf /bin/ls).

Se invece avete scaricato un nuovo pacchetto rpm, ma non sapete esattamente cosa sia, si



può vedere che informazioni ci hanno attaccato gli autori, con il comando

```
rpm -qpi nomeCompletoDelPacchetto
```

Eseguendo il comando sul pacchetto `mplayer-common`, che abbiamo scaricato ed installato nella scorsa lezione, si ottiene il seguente risultato:

```
Name      : mplayer-common          Relocations: (not relocateable)
Version   : 0.90rc3                Vendor: The MPlayer Project
Release   : 2                      Build Date: mer 22 gen 2003
23:41:55 CET
Install date: (not installed)      Build Host:
rathann.rangers.eu.org
Group     : Applications/Multimedia Source RPM: mplayer-
0.90rc3-2.src.rpm
Size      : 2375944                License: GPL
Signature : DSA/SHA1, gio 01 gen 1970 01:00:00 CET, Key ID 0000000000000000
Packager  : Dominik Mierzejewski <dominik@rangers.eu.org>
URL       : http://www.mplayerhq.hu
```



Summary : Configuration files and documentation for mplayer.

Description :

Configuration files, man page and HTML documentation for mplayer.

Notate che ci sono numerose informazioni legate ad un pacchetto, che permettono di caratterizzarlo a pieno per la sua versione, per il suo tipo (in questo caso Multimedia), e per chi l'ha prodotto. Ci sono inoltre informazioni su dove trovare il codice sorgente (obbligatorie, se il software è "free"), e descrizioni del sistema in cui è stato compilato, per capire eventualmente cosa non va.

Richiedendo poi la pagina del manuale relativa a rpm (comando 'man rpm'), è possibile vedere le innumerevoli opzioni che si hanno a disposizione per la gestione dei pacchetti, anche se raramente se ne usano più di quante riportate in questa lezione.

Che fare se invece si vuole eliminare un programma che non è stato installato con il metodo rpm?

Le cose si complicano notevolmente, perché non c'è un metodo standard. Alcuni programmi, come Netscape 7 o Acrobat Reader si installano in una directory separata da tutto il resto (ad esempio /usr/local/Acrobat). E' dunque sufficiente eliminare questa per aver risolto semplicemente il problema.

Altri programmi di solito vengono installati nella directory /usr/local come base, andandosi a distribuire nelle sottodirectory:

- bin per quanto riguarda gli eseguibili
- share per quanto riguarda i dati (icone, file contenenti i messaggi visualizzati, etc)
- lib per quanto riguarda le librerie di supporto
- etc per quanto la configurazione

A meno che voi non sappiate ESATTAMENTE di quali file è composto il programma che volete disinstallare, vi sconsiglio di provare a farlo manualmente, perché rischiate di compromettere il funzionamento di tutto il sistema!

Dopo questa lezione, sapete come gestire i pacchetti nel vostro sistema: nella prossima esercitazione avrete modo di installare alcune applicazioni davvero utili che mancano all'appello, mentre nelle prossime lezioni impareremo altri piccoli trucchi per rendere il nostro sistema davvero funzionale.