Corso Linux ARCES

Lezione 4:
Gestione dei pacchetti

Metodi di archiviazione e compressione

Archiviazione:

- Un archivio è un file contenente al suo interno altri files (un pó come una directory). Sotto Linux si possono creare o aprire gli archivi mediante il comando tar;
- Tuttavia praticamente tutti i filemanager possiedono delle utility grafiche in grado di gestire gli archivi (ad esempio ark);

Esempio:

```
tar -cf archivio.tar foo bar      # Crea archivio.tar dai file foo e bar.
tar -xf archive.tar               # Estrae tutti i file da archivio.tar.
```

Metodi di archiviazione e compressione

Compressione:

- Esistono diversi tool capaci di comprimere un file (di qualunque tipo) alcuni grafici altri testuali. Tra questi ricordiamo: gzip (.gz); bunzip (.bz); bzip (.bz); zip (.zip) ecc. Questo comporta la necessità di utilizzare il programma adatto per ogni tipo specifico di file compresso;

Esempio:

gzip -vd <file.gz> scompatta file.gz
gzip -v9 <file> comprime file creando file.gz

Installazione dei pacchetti tipo tar ball

- Molti packages sono disponibili come archivi compressi con estensione tar.gz o simile. Questo genere di pacchetto viene denominato tar ball;
- Per installare questi pacchetti è necessario compiere quindi alcune operazioni preliminari. La più importante di queste è la decompressione/de-archiviazione;
- Successivamente si può passare all'installazione vera è propria del pacchetto sul sistema;

Esempio:

gzip -vd <file.tar.gz>
tar -xvf <file.tar>
tar -xvzf <file.tar.gz>

Installazione di pacchetti tipo tar ball (2)

- Una volta decompresso l'archivio prima di installare il pacchetto, soprattutto se non si è sicuri, è bene leggere i files INSTALL e README presenti nel 99,9 % dei casi. Questi files riescono in gran parte dei casi a far luce nel mare delle tenebre. Successivamente passati in modalità amministratore bisogna eseguire i seguenti comandi:

1) ./configure
2) make
3) make install

Red hat Package Manager

- Il progetto RPM nasce sotto la supervisione di Red Hat (che deve ad esso parte del suo successo) anche se poi è stato utilizzato da moltissime altre distribuzioni (SuSE, Mandrake, ecc);
- La causa del successo di RPM è duplice: Innanzitutto le aziende evitano in questa maniera di dare “ in pasto ” agli utenti i loro prodotti. Inoltre RPM consente di installare software senza aver bisogno di grandissime conoscenze poiché automatizza al massimo il processo di installazione evitando all'utente di commettere errori pur tuttavia garantendo un'ottima qualità del prodotto finito;

RPM (2)

- RPM è quindi un programma che consente di:

0) Creare pacchetti;
1) Installare;
2) Disinstallare;
3) Aggiornare;
4) Interrogare;
5) Verificare;

NOTA: Per avere QUALUNQUE informazione su rpm è possibile usare i comandi:

help rpm; man rpm; rpm --help

RPM – Installazione

- Ogni pacchetto RPM ha un nome che segue una sintassi abbastanza precisa:

foo-1.0-1.i386.rpm

nome-versione-release-architettura

- Per poter installare un pachetto RPM bisogna passare in modalità super utente o amministratore (root):

su - root

- Solo allora si può digitare il comando:

rpm -Uvh <file.rpm>

RPM – Installazione (2)

- Se si installano packages del kernel invece:

rpm -ivh <file.rpm>

Eventuali conflitti:

- Package già installato:

per forzare: rpm -ivh --replacepkgs <file.rpm>

- Conflitto tra files:

per ignorare: rpm -ivh --replacefiles <file.rpm>

RPM – Installazione (3)

- Dipendenze irrisolte:

per risolvere: rpm -ivh <file1.rpm> <file2.rpm>

rpm -q --redhatprovides <file2.rpm>

per forzare: rpm -ivh --nodep <file.rpm>

RPM - Disinstallazione

- Per disinstallare un determinato pacchetto RPM bisogna digitare il comando:

rpm -e <nome rpm>

- Potrebbero sorgere dei conflitti dovuti a delle dipendenze tra packages. In questo caso:

per forzare: rpm -e --nodeps <nome rpm>

RPM - Aggiornamento

- Quando si fa l'aggiornamento di un pacchetto RPM disinstalla automaticamente la versione precedente e la sostituisce con quella nuova. In pratica fa quello che si dice un upgrade intelligente;
- L'upgrade è una combinazione di disinstallazione ed installazione. È quindi possibile avere dei fallimenti del processo in ogniuna di queste fasi;
- Per aggiornare un pacchetto bisogna quindi digitare il comando:

rpm -Uvh <file.rpm>

RPM – Aggiornamento (2)

- Upgrade di un pacchetto con conservazione del vecchio:

  rpm -Uvh --oldpackage <file.rpm>

- Il refresh non installa se non è presente una versione precedente nel sistema:

  rpm -Fvh <file.rpm>

RPM - Interrogazione

- Interrogare il sistema RPM serve per controllare se è presente un dato pacchetto ed eventualmente accedere ad un database contenente informazioni ad esso relative;
- Per interrogare il database RPM basta digitare:

rpm -q (nome, versione, release)

Opzioni:

-a	tutti i packages;
-f <file>	(path completo);
-p <packagefile>	

RPM - Verifica

- La verifica serve a confrontare le informazioni del pacchetto installato per rilevare eventuali errori;
- Per verificare un pacchetto basta digitare:

rpm -V <pacchetto>

- Verifica di tutti i pacchetti:

rpm -Va

- Verifica mediante confronto:

rpm -Vp <package noto>

- Se la verifica ha esito positivo nessun output

Controllare la firma RPM

- Si possono effettuare due tipi di verifica sui packages RPM: Verificare che i files non siano corrotti e verificare l'affidabilità dell'autore del package;
- Questo può essere fatto avvalendosi dell'algoritmo md5 che è capace di confrontare lo stato iniziale e finale di un processo (es. download) allo scopo di individuare eventuali corruzioni nei files;
- Inoltre per controllare l'affidabilità dell'autore di un pacchetto ci si può avvalere del sistema di chiavi pubblico/private GPG;

Controllare la firma RPM (2)

- Verificare l'integrità del file:

```
rpm -k --nogpg  <pacchetto>
```

- Verificare l'affidabilità dell'autore:

```
rpm -k  <pacchetto>
```

- Importare chiavi pubbliche:

```
rpm -qa gpg-pubkey
```

- informazioni su una chiave:

```
rpm -qi  gpg-pubkey-<chiave>
```

YUM & apt-get

- Esistono diversi (anche troppi) tools (grafici e non) che permettono di “semplificare” la gestione dei pacchetti e delle operazioni ad essa connessa;
- Questi programmi sono spesso customizzabili secondo le esigenze dell'utente in modo da ottimizzarne le prestazioni in relazione alle necessità;
- Noi ci soffermeremo su **Yellow dog Updater Modified** e **apt-get**;

Yellow dog Updater Modified

- Nato presso il LUG (Linux User Group) della Duke University è un' utility che consente di aggiornare, installare e rimuovere pacchetti RPM con l'aggiunta (e non è cosa da poco) della capacità di risolvere automaticamente le dipendenze;
- Ha una sintassi estremamente semplice che risparmia molte fatiche all'utente (per esempio quella di ricordare la versione ESATTA del pacchetto da installare). Dispone inoltre di una propria lista (configurabile) di repositories;

yum [opzioni] [comando] <pacchetto>

apt-get

- Originariamente nato come package manager per Debian APT è stato esportato su numerose altre distribuzioni (SuSE, Fedora, ecc). Anche apt-get consente di risolvere automaticamente le dipendenze;
- Ha una sintassi molto semplice per alcuni versi simile a YUM ;

apt-get [opzioni] [comando] <pacchetto>

Compilatori GNU

- Un compilatore è un programma che converte un file scritto in un linguaggio in un file eseguibile in linguaggio macchina.
- All'inizio i compilatori erano scritti appositamente per ogni macchina ma nel '78 Steve Johnson scrisse il primo compilatore portabile del linguaggio “C”;
- Il compilatore GNU C per eccellenza è il gcc. gcc funziona anche come compilatore per: C, C++ , Objective-C; Fortran, Java e Ada;

Links utili & bibliografia

- http://www.linuxquestions.org
- http://www.rpm.org/
- http://www.debian.org/doc/manuals/apt-howto/index.en.
- http://linux.duke.edu/projects/yum/
- http://gcc.gnu.org/
- http://cm.bell-labs.com/cm/cs/who/dmr/chist.html
- http://www.oltrelinux.org
- http://www.google.com