



ADMINISTRATION DES SYSTÈMES ET RÉSEAUX

INSTALLATION DE LOGICIELS SOUS LINUX

Auteur: Bernard GIACOMONI
Autoentreprise GIACOMONI Bernard
Création de logiciels, Conseil et formations

Version	Date	Objet
1.0	10/05/2019	Version initiale

Table des matières

I. INTRODUCTION :	3
II. LES PACKAGES RPM (RED HAT):	3
II.1. PRÉSENTATION.....	3
II.2. LES COMMANDES LINUX RPM :	4
II.2.1. INSTALLATION:.....	4
II.2.2. SUPPRESSION:.....	4
II.2.3. MISE À JOUR:.....	4
II.2.4. INTERROGATION :	4
II.2.5. VÉRIFICATION:.....	4
III. LES PACKAGES DEB.....	6
III.1. LES COMMANDES DPKG:.....	6
III.2. LES COMMANDES APT :	7
IV. INSTALLATION CLASSIQUE A PARTIR D'ARCHIVES :	10
IV.1. PREMIÈRE PHASE : DÉCOMPRESSION :	10
IV.2. COMPILATION :	11
IV.3. INSTALLATION :	11
V. EXÉCUTION :	11

I. INTRODUCTION :

Il existe plusieurs façons de récupérer les fichiers d'installation d'une application fonctionnant sous Linux :

- Sous forme de paquetage RPM. Il s'agit d'un fichier portant l'extension .rpm permettant une installation automatique, pour les distributions Linux telles que RedHat ou Mandrake.
- Sous forme de paquetage DEB. Il s'agit d'un fichier portant l'extension .deb permettant une installation automatique, pour les distributions Linux Debian.
- Sous forme d'archive. Il s'agit de fichiers compressés comportant l'extension .tar.gz, .gz ou .tar.bz2

Dans le dernier cas il s'agit généralement des sources du logiciel. Il vous faut donc compiler le programme sur votre machine afin de rendre son exécution possible.

II. LES PACKAGES RPM (RED HAT):

II.1. PRÉSENTATION

RPM (Red Hat Package Manager) est, comme son nom l'indique, un outil de gestion de packages (Un package étant pour linux un ensemble de fichiers constituant une application. L'outil permet d'installer des applications, et de les supprimer plus facilement facilement qu'en employant des commandes « de base ».

RPM installe directement les « bons » fichiers dans les « bons » répertoires. De plus, RPM propose quatre fonctions supplémentaires :

1. Mise à jour les packages: il n'est pas nécessaire de désinstaller la version précédente avant d'installer la nouvelle : RPM permet une mise à jour automatique des packages.
2. Interrogation: RPM permet de connaître le contenu d'un package, ou de trouver à quel package appartient un fichier.
3. Vérification du système: RPM permet de vérifier l'intégrité d'un package (déterminer si tous les fichiers que contenait le package sont bien installés dans le système).
4. Accès aux sources: RPM met à disposition des utilisateurs d'un package les sources, ainsi que les instructions nécessaires à la compilation au cas où des utilisateurs voudraient modifier les fichiers du package.

II.2.LES COMMANDES LINUX RPM :

Voici les commandes des 5 fonctionnalités principales :

II.2.1.INSTALLATION:

pour installer le package "mpg123-0_59p-1_i386.rpm", saisir la commande :

```
rpm -ivh mpg123-0_59p-1_i386.rpm
```

Si le package a déjà été installé précédemment le système vous affichera le message

```
package mpg123-0_59p-1 is already installed
```

(L'option -replacepks permet de forcer l'installation)

II.2.2.SUPPRESSION:

pour supprimer un package il suffit de saisir la commande :

```
rpm -e mpg123 (mpg123 représente le nom du package à désinstaller)
```

II.2.3.MISE À JOUR:

Pour mettre à jour un package (ce qui part du principe qu'une version précédente du package a déjà été installée...) il faut saisir la commande :

```
rpm -Uvh mpg123-0_59p-1_i386.rpm
```

II.2.4.INTERROGATION :

Il existe plusieurs commandes visant à interroger un ou plusieurs packages selon divers critères. Les voici :

Pour connaître des détails sur un package particulier :

```
rpm -q mpg123-0_59p-1_i386.rpm
```

Pour interroger tous les packages existants :

```
rpm -qa
```

Pour interroger les packages contenant un fichier particulier :

```
rpm -f NomDuFichier
```

Pour avoir des informations sur un package :

```
rpm -p NomDuPackage
```

II.2.5.VÉRIFICATION:

Cette option permet de comparer les différences entre le package installé et le package tel qu'il devrait être. Les commandes sont :

Pour vérifier vis-à-vis d'un fichier particulier :

```
rpm -Vf NomDuFichier
```

Pour vérifier tous les packages :

```
rpm -Va
```

Pour vérifier un package particulier :

`rpm -Vp NomDuPackage`

Lorsqu'il n'y a pas d'erreur, le système n'affiche rien. Si par contre une erreur a été détectée dans le package installé, le système renvoie une chaîne de huit caractères indiquant le type d'erreur. Voici la signification des caractères :

- D: Périphérique
- G: Groupe
- L: Lien symbolique
- M: Mode (Permissions)
- T: date du fichier
- S: Taille du fichier
- U: Utilisateur

III.LES PACKAGES DEB

III.1.LES COMMANDES DPKG:

La distribution Debian propose un outil de gestion de package permettant d'automatiser l'installation, la configuration et la mise à jour des logiciels installés par ce biais. Les packages Debian sont ainsi des fichiers portant l'extension .deb et pouvant être installés manuellement grâce à la commande suivante :

```
dpkg -i nom_du_package.deb
```

REMARQUE :

> sudo dpkg --get-selections permet d'afficher la liste des paquets installés.

III.2.LES COMMANDES APT :

De plus un système complémentaire a été mis au point afin de gérer les dépendances entre les différents packages et de permettre de les mettre à jour par simple téléchargement: il s'agit du système apt (Advanced Packaging Tool). Ce système fonctionne à partir d'une liste de sources (emplacements auxquels des paquetages sont disponibles) présente dans le fichier /etc/apt/sources.list.

Installation:

pour installer le package "nom_du_package" il suffira de taper :

```
apt-get install nom_du_package
```

ou bien, après avoir récupéré manuellement un package spécifique (avec l'extension .deb)

```
dpkg --install nom_du_package.deb ou dpkg -i nom_du_package.deb
```

En cas de problème de dépendance il est possible de forcer l'installation avec la commande suivante :

```
apt-get install -f nom_du_package
```

Simulation d'installation:

il peut être utile de simuler l'installation du package avant de procéder à son installation afin de connaître exactement les paquetages dépendants qui vont être installés et estimer les problèmes éventuels :

```
apt-get install -s nom_du_package
```

Suppression:

pour supprimer un package il suffit de saisir la commande :

```
apt-get remove nom_du_package
```

Cette commande supprime le paquet mais conserve les fichiers de configuration.

Supprimer également les fichiers de configuration :

il suffit d'utiliser la commande suivante :

```
apt-get remove --purge nom_du_package
```

Reconfiguration:

pour reconfigurer un package il suffit de saisir la commande:

```
dpkg --reconfigure nom_du_package
```

Mise à jour de la liste des packages:

Pour mettre à jour la liste des packages il suffit de saisir la commande :

```
apt-get update
```

Mise à jour des packages:

La mise à jour des packages eux-mêmes se fait grâce à la commande suivante :

```
apt-get upgrade
```

Il est conseillé de lancer cette commande avec le commutateur -u afin de connaître la liste exacte des paquetages qui vont être mis à jour :

```
apt-get -u upgrade
```

La commande suivant permet également de mettre à jour les paquets en préservant mieux les dépendances afin de conserver un système plus cohérent :

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

Nettoyage:

à chaque installation d'un logiciel, le système APT charge le paquetage concerné et le stocke dans un référentiel local. Au bout d'un certain temps la liste des paquetages conservés peut devenir importante et gourmande en espace disque, c'est la raison pour laquelle il peut être utile de faire le ménage parmi ces fichiers. La commande clean permet de supprimer tous les paquetages n'étant pas verrouillés, ce qui signifie que le paquetage devra être à nouveau téléchargé s'il doit être reconfiguré :

```
apt-get clean
```

La commande autoclean permet de supprimer tous les paquetages n'étant plus disponibles en téléchargement (et donc a priori obsolètes :

```
apt-get autoclean
```

Interrogation :

Il existe plusieurs commandes visant à interroger un ou plusieurs packages selon divers critères. Les voici :

Pour connaître la liste des packages contenant un mot clé particulier :

```
apt-cache search mot_cle
```

Pour obtenir plus d'information sur un paquetage donné :

```
apt-cache show nom_du_package
```

Pour afficher les dépendances d'un paquetage :

```
apt-cache depends nom_du_package
```

Pour connaître le paquetage auquel un fichier appartient :

```
apt-file search nom_du_fichier
```


Pour connaître la liste des fichiers d'un paquetage :

```
apt-file list nom_du_package
```

Le logiciel `deborphan` permet d'indiquer le nom de toutes les bibliothèques qui ne sont plus utilisées sur un système `debian`. Ces bibliothèques sont appelées orphelines car plus aucun paquet ne dépend d'elles. Cet outil est notamment très pratique pour nettoyer les paquets inutiles.

Enfin, il existe des outils graphiques qui permettent de simplifier la gestion des paquetages:

- `Aptitude`
- `Synaptic`
- `Gnome-apt`
- `Wajig`

IV.INSTALLATION CLASSIQUE A PARTIR D'ARCHIVES :

IV.1.PREMIÈRE PHASE : DÉCOMPRESSION :

- Les programmes sont la plupart du temps fournis compressés, c'est-à-dire sous un format plus compact permettant de réduire la taille du programme, notamment pour faciliter son téléchargement, c'est le programme GZIP qui permet ce compactage.
- Plusieurs fichiers peuvent aussi être simplement regroupés en une seule archive pour faciliter leur transport, c'est le rôle du programme TAR.
- Enfin, les fichiers peuvent être compressés et regroupés en une archive portant l'extension .tar.gz.

Les données compressées existent sous plusieurs formats :

Format Tar:

les données sous ce format portent l'extension .tar. Pour décompresser ce type de données il faut saisir en ligne de commande :

```
tar xvf nom_du_fichier.tar
```

Format Gzip:

les fichiers compressés en Gzip possèdent l'extension .gz. Pour décompresser ces fichiers il faut saisir en ligne de commande :

```
gunzip nom_du_fichier.gz
```

Format Bzip2:

les fichiers compressés en Bzip2 possèdent l'extension .bz2. Pour décompresser ces fichiers il faut saisir en ligne de commande :

```
bzip2 -d nom_du_fichier.bz2
```

Format Tar/Gzip (on parle généralement de Tarball):

Les données compressées en TAR et en GZIP portent l'extension .tar.gz. Elles peuvent être décompressées successivement par les deux moyens énoncés ci-dessus ou à l'aide de la commande :

```
tar zxvf nom_du_fichier.tar.gz
```

Format Tar/BZip2:

Les données compressées en Tar et en Bz2 portent l'extension .tar.bz2. Elles peuvent être décompressées successivement par les deux moyens énoncés ci-dessus ou par la commande :

```
tar jxvf nom_du_fichier.tar.bz2
```

Format Compress:

les fichiers archivés par le programme Compress portent l'extension .Z. Ils peuvent être décompressés par le programme **Uncompress**.

IV.2.COMPILATION :

Lorsque le programme est une archive (et non un RPM) il est généralement nécessaire de compiler les sources.

Bien que la compilation se fasse pratiquement toujours de la même façon pour la majorité des applications, il est vivement conseillé de lire le fichier INSTALL ou README pour connaître exactement les commandes nécessaires à la compilation du programme.

Dans un premier temps il faut exécuter la commande :

```
./configure
```

Elle permet d'adapter le programme à la machine courante. Les options de la commande configure sont :

- Pour obtenir de l'aide : `./configure --help`
- Pour installer l'application dans un répertoire spécifique : `./configure --prefix="repertoire"`

Dans un second temps il faut compiler le programme, grâce à la directive :

```
make
```

Cette commande peut prendre du temps en fonction de votre configuration et du logiciel à compiler. En cas de problème, la compilation s'arrêtera en renvoyant un message d'erreur. Si tout se passe normalement, après affichage de nombreuses lignes de traces, le programme reviendra à la ligne de commande.

IV.3.INSTALLATION :

L'installation de l'application doit s'effectuer en tant que super utilisateur (root). Pour installer l'application, la syntaxe suivante est généralement utilisée :

```
make install
```

En cas d'échec lors de la compilation, il faut nettoyer les fichiers makefile ayant été créés automatiquement lors de la compilation avant de recommencer une nouvelle compilation grâce à la commande :

```
make clean
```

V.EXÉCUTION :

Le programme exécutable se trouve soit dans le répertoire spécifié lors de l'installation, soit dans : `/usr/local/nom_du_programme`.