ROYAUME DU MAROC

مكتب التكوين المهني وإنعاش الشغل

Office de la Formation Professionnelle et de la Promotion du Travail

Installation d'applications sous Gnu/Linux

OFPPT

DIRECTION **R**ECHERCHE ET **I**NGENIERIE DE **F**ORMATION

SECTEUR NTIC

Sommaire

1. Les diffÉrents types d'archives

Basiquement il existe plusieurs faÃ§ons de rÉcupÉrer 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 second 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.

2. Les packages RPM

RPM (Red Hat Package Manager) est, comme son nom l'indique (en anglais), un outil de gestion de packages (ensemble de fichiers constituant une application) pour Linux. Il permet d'installer des applications, ainsi que les supprimer facilement.
GrÃ¢ce Ã une commande simple, les fichiers s'installent directement dans les "bons" rÉpertoires.
RPM a quatre utilitÉs supplÉmentaires:

* Mettre Ã jour les packages: Lorsque vous rÉcuperez une nouvelle version d'un package, il n'est pas nÉcessaire de dÉsinstaller la version prÉcÉdente avant d'installer la nouvelle. En effet RPM permet une mise Ã jour automatique des packages!
* Interrogation: RPM permet effectivement de connaÃ®tre le contenu d'un package, mais il permet aussi de savoir Ã quel package appartient un fichier.
* VÉrification du systÈme: RPM permet de vÉrifier l'intÉgritÉ d'un package (c'est-Ã -dire de dÉterminer si tous les fichiers que contenait le package sont bien installÉs dans le systÈme.
* AccÈs aux sources: RPM met Ã disposition des utilisateurs d'un package les sources, ainsi que les instructions nÉcessaires Ã la compilation au cas oÃ¹ des utilisateurs voudraient modifier les fichiers du package.

Voici les commandes des 5 fonctionnalitÉs principales:

* Installation: pour installer le package "mpg123-0_59p-1_i386.rpm" il suffira de taper:

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

Si le package a dÉjÃ ÉtÉ installÉ prÉcÉdemment le systÈme vous affichera le message
package mpg123-0_59p-1 is already installed
(L'option -replacepkgs permet de forcer l'installation)
* Suppression: pour supprimer un package il suffit de taper

rpm -e mpg123

(OÃ¹ mpg123 reprÉsente le nom du package Ã dÉsinstaller, ce sera un autre nom pour vous...)
* Mise Ã jour: Pour mettre Ã jour un package (ce qui part du principe qu'une version prÉcÉdente du package a dÉjÃ ÉtÉ installÉe...) il faut taper la commande:

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

* Interrogation : Il existe plusieurs commandes visant Ã interroger un ou plusieurs packages selon divers critÈres. Les voici:
o Pour connaÃ®tre des dÉtails sur un package particulier:

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

o Pour interroger tous les packages existants:

rpm -qa

o Pour interroger les packages contenant un fichier particulier:

rpm -f NomDuFichier

o Pour avoir des informations sur un package:

rpm -p NomDuPackage

* VÉrification: Cette option permet de comparer les diffÉrences entre le package installÉ et le package tel qu'il devrait Ãªtre. Les commandes sont
o Pour vÉrifier vis-Ã -vis d'un fichier particulier:

rpm -Vf NomDuFichier

o Pour vÉrifier tous les packages:

rpm -Va

o 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 chaÃ®ne de huit caractÈres indiquant le type d'erreur. Voici la signification des caractÈres
o D: PÉriphÉrique
o G: Groupe
o L: Lien symbolique
o M: Mode (Permissions)

o T: date du fichier
o S: Taille du fichier
o U: Utilisateur

3. **Les packages DEB**

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

dpkg -i nom_du_package.deb

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 Ã jour par simple tÉlÉchargement: il s'agit du systÈme apt (Advanced Packaging Tool). Ce systÈme fonctionne Ã partir d'une liste de sources (emplacements auxquels des paquetages sont disponibles) prÉsente dans le fichier /etc/apt/sources.list.

Voici les commandes des fonctionnalitÉs principales:

* 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

En cas de problÈme de dÉpendance il est possible (mais dÉconseillÉ, sauf en connaissance de cause) de forcer l'installation avec la commande suivante :

apt-get install -f nom_du_package

* Simulation d'installation: il peut Ãªtre utile de simuler l'installation du package avant de procÉder Ã son installation afin de connaÃ®tre exactement les paquetages dÉpendants qui vont Ãªtre installÉs et estimer les problÈmes Éventuels :

apt-get install -s nom_du_package

* Suppression: pour supprimer un package il suffit de taper

apt-get remove nom_du_package

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

dpkg --reconfigure nom_du_package

* Mise Ã jour de la liste: Pour mettre Ã jour la liste des packages il faut taper la commande:

apt-get update

* Mise Ã jour des packages: La mise Ã jour des packages eux-mÃªmes se fait grÃ¢ce Ã la commande suivante :

apt-get upgrade

Il est conseillÉ de lancer cette commande avec le commutateur -u afin de connaÃ®tre la liste exacte des paquetages qui vont Ãªtre mis Ã jour :

apt-get -u upgrade

La commande suivant permet Également de mettre Ã jour les paquets en prÉservant mieux les dÉpendances afin de conserver un systÈme plus cohÉrent :

apt-get dist-upgrade

* Nettoyage: Ã 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 consrvÉs peut devenir importante et gourmande en espace disque, c'est la raison pour laquelle il peut Ãªtre utile de faire le mÉnage parmi ces fichiers. La commande clean permet de supprimer tous les paquetages n'Étant pas verrouillÉ, ce qui signifie que le paquetage devra Ãªtre Ã nouveau tÉlÉchargÉ s'il doit Ãªtre 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 Ã interroger un ou plusieurs packages selon divers critÈres. Les voici:
o Pour connaÃ®tre la liste des packages contenant un mot clÉ particulier :

apt-cache search mot_cle

o Pour obtenir plus d'information sur un paquetage donnÉ :

apt-cache show nom_du_package

o Pour afficher les dÉpendances d'un paquetage :

apt-cache depends nom_du_package

o Pour connaÃ®tre le paquetage auquel un fichier appartient :

apt-file search nom_du_fichier

o Pour connaÃ®tre la lite des fichiers d'un paquetage :

apt-file list nom_du_package

Le logiciel deborphan permet d'indiquer le nom de toutes les librairies qui ne sont plus utilisÉes sur un systÈme debian. Ces librairies sont appelÉes orphelines car plus aucun paquet ne dÉpend d'elles. Cet outil est notamment trÈs pratique pour nettoyer les paquets inutiles.

Enfin, il est Ã noter l'existe d'interfaces graphiques permettant de simplifier la gestion des paquetages :

* Aptitude
* Synaptid
* Gnome-apt
* Wajig

4. DÉcompresser des archives

Les programmes sont la plupart du temps fournis compressÉs, c'est-Ã -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 Ãªtre simplement regroupÉs en une seule archive pour faciliter leur transport, c'est le rÃ´le du programme TAR. Enfin, les fichiers peuvent Ãªtre compressÉs et regroupÉ en une archive portant l'extension .tar.gz.

Les donnÉes compressÉes existent sous plusieurs formats:

* Tar: les donnÉes sous ce format portent l'extension .tar
pour dÉcompresser ce type de donnÉes il faut taper en ligne de commande:

tar xvf nom_du_fichier.tar

* Gzip: les fichiers compressÉes en Gzip possÈdent l'extension .gz
pour dÉcompresser ces fichiers il faut taper en ligne de commande:

gunzip nom_du_fichier.gz

* Bzip2: les fichiers compressÉes en Bzip2 possÈdent l'extension .bz2
pour dÉcompresser ces fichiers il faut taper en ligne de commande:

bzip2 -d nom_du_fichier.bz2

* Tar/GZip (on parle gÉnÉralement de Tarball): les donnÉes compressÉes en TAR et en GZIP portent l'extension .tar.gz
Elles peuvent Ãªtre dÉcompressÉes successivement par les deux moyens ÉnoncÉs ci-dessus ou Ã l'aide de la commande:

tar zxvf nom_du_fichier.tar.gz

* Tar/BZip2: les donnÉes compressÉes en Tar et en Bz2 portent l'extension .tar.bz2
Elles peuvent Ãªtre dÉcompressÉes successivement par les deux moyens ÉnoncÉs ci-dessus ou Ã l'aide de la commande:

tar jxvf nom_du_fichier.tar.bz2

* Compress: les fichiers archivÉs par le programme Compress portent l'extension .Z, ils peuvent Ãªtre dÉcompressÉs par le programme Uncompress.

Compiler le programme

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 mÃªme faÃ§on pour la majoritÉ des applications, il est vivement conseillÉ de lire le fichier INSTALL ou README pour connaÃ®tre exactement les commandes nÉcessaires Ã la compilation du programme.

Dans un premier temps il faut exÉcuter la commande:

./configure

Cette commande est un script permettant d'adapter le programme Ã la machine courante. Les options de 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, grÃ¢ce Ã la directive

make

Cette commande peut prendre plusieurs minutes/heures selon votre configuration et le logiciel Ã compiler. En cas de probleme, la compilation s'arretera en vous renvoyant un message d'erreur. Si tout se passe normalement de nombreuses lignes vont Ãªtre affichÉes, puis le programme reviendra Ã la ligne de commande.

AprÈs avoir compilÉ l'application, il est possible de l'installer

L'installation de l'application doit s'effectuer en tant que superutilisateur (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. Cela se fait grÃ¢ce Ã la commande:

make clean

Une fois toutes ces Étapes franchies, votre programme devrait Ãªtre exÉcutable et se trouver soit dans le rÉpertoire spÉcifiÉ lors de l'installation, ou bien gÉnÉralement dans /usr/local/nom_du_programme.