

CHAPITRE 4 : La gestion des packages logiciels Sous Linux

Mohammed SABER

Département Électronique, Informatique et Télécommunications
École Nationale des Sciences Appliquées "ENSA"
Université Mohammed Premier OUJDA

Année Universitaire : 2018-2019

Plan de chapitre

1 Introduction

2 Les Archives

3 Les packages Debian

Plan de chapitre

1 Introduction

2 Les Archives

3 Les packages Debian

Introduction

- Comme la plupart des systèmes d'exploitation, Linux permet l'installation de nouvelles applications mais aussi leur mise à jour et leur désinstallation.
- On trouve généralement les logiciels sous deux formes :
 - Une archive contenant les sources du logiciel ;
 - Un package contenant le programme déjà compilé ainsi que les scripts nécessaires à son installation.

Plan de chapitre

1 Introduction

2 Les Archives

3 Les packages Debian

Les étapes d'installation

Étape 1 : Désarchivage des fichiers sources

Le programme `tar` permet d'extraire et de décompresser les fichiers présents dans l'archive au format `.tar.gz` (le détail de cette commande voir chapitre sauvegarde).

```
root@Nom_machine :~# ls
Hello-1.3.tar.gz
root@Nom_machine :~# tar -xvzf Hello-1.3.tar.gz
```

Étape 2 : lecture des instructions de compilation et d'installation

Il existe dans la plupart des archives certains fichiers remarquables :

- **README** : il contient notamment, des informations sur les fonctionnalités de l'application, ses auteurs et leurs coordonnées.
- **COPYING «copyright»** : copie du texte de la licence sous laquelle est développé le logiciel ; « **GPL** ».
- **INSTALL** : instructions d'installation qu'il suffit généralement de suivre pas à pas.

Les Archives

- Une très grande majorité des applications utilisées sous Linux étant sous licence GPL et assimilées, les fichiers sources de la plupart des programmes sont disponibles sous forme d'archives compressées.
- Ces fichiers archives portent habituellement l'extension `.tar.gz` ou `.tgz` dans le cas de fichiers à archiver à l'aide de la commande `tar` et compressés avec `gzip`, et l'extension `.tar.Z` lorsque cette archive est compressée avec `compress`.
- Ces fichiers contiennent tous les composants de l'application : les exécutables, les bibliothèques, les fichiers de configuration, la documentation, des fichiers de données, des exemples, ...

Les étapes d'installation

Étape 3 : configuration de la compilation

- Le fichier `configure` présent dans l'archive se charge de configurer la compilation en fonction de l'environnement sur lequel il est exécuté ; il détermine par exemple la distribution utilisée, les ressources matérielles disponibles, les logiciels et bibliothèques déjà installés.

```
root@Nom_machine :~Archive# ./configure
Checking ...
...
Creating Makefile
```

- Ce script crée en fait un fichier de commandes nommé `Makefile` utilisé par l'outil de compilation standard `make`.

Étape 4 : Compilation

Grâce à la commande `make` qui va rechercher les noms des différents fichiers sources et les options de compilation dans le fichier `Makefile` généré précédemment, le programme va être compilé.

Les étapes d'installation

Étape 5 : Installation des fichiers compilés

Cette même commande `make` accepte habituellement l'argument `install`, ce qui a pour effet de copier les fichiers compilés précédemment dans l'arborescence du système.

```
root@Nom_machine :~Archive# make install
....
```

Étape 6 : configuration

Des fichiers de configuration standard sont généralement fournis et installés dans le même répertoire du programme ou dans `/etc`.

Les packages Debian

- Les packages Debian, utilisés par la distribution du même nom, portent l'extension « `.deb` » pour les paquets binaires et « `.dsc` » pour les paquets sources.
- Les fichiers `.deb` contiennent toutes les informations nécessaires à l'installation/désinstallation de packages logiciels sur le système, à savoir :
 - Le nom, la version et la description du package ;
 - Les dépendances entre packages ;
 - Les programmes de pré, postinstallation et désinstallation ;
 - Les fichiers du logiciel.
 - Le nom typique d'un fichier Debian est : `foo-1.3-2.deb` (Désigne la deuxième version (2) du package (`.deb`) de la version (1.3) du logiciel `foo`).

Plan de chapitre

- 1 Introduction
- 2 Les Archives
- 3 Les packages Debian

Gestion de package DPKG

- La commande `dpkg` est chargée de l'installation, la création, la suppression et la gestion des paquets Debian.
- La commande de gestion des package bas niveau en mode ligne de commande.
- La base de données `dpkg` est généralement placée dans `/var/lib/dpkg`.
- Les fichiers qui y sont présents sont au format texte.
- Cependant n'écrivez pas les fichiers à la main. Le fichier `/var/lib/dpkg/status` contient l'intégralité des packages connus par `dpkg` avec leur état.

Gestion de packages DPKG

- Afficher l'aide en ligne (-h ou --help) :

```
root@Nom_machine :~# dpkg -h
```

- Installer un package (-i ou --install) :

```
root@Nom_machine :~# dpkg -i package.deb
```

- Supprimer un package (r ou --remove) :

```
root@Nom_machine :~# dpkg -r package
```

- Lister les packages installés (-l ou --list) :

```
root@Nom_machine :~# dpkg -l package
```

- Visualiser les caractéristiques d'un package (-s ou --status) :

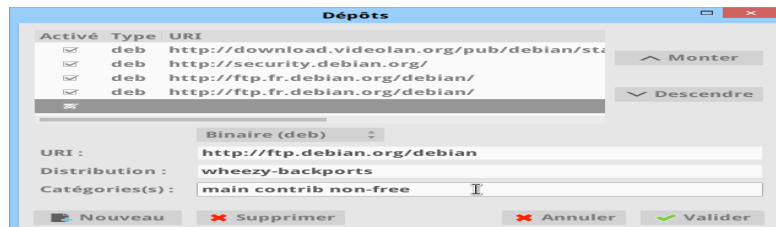
```
root@Nom_machine :~# dpkg -s package
```

- Lister les fichiers composants d'un package :

```
root@Nom_machine :~# dpkg -T package
```

Gestion de package APT

- Les dépôts dans le fichier sources.list
- Un dépôt contient un ensemble de packages qui dépendent soit les uns des autres, soit d'autres packages en provenance d'autres dépôts.
- APT peut gérer plusieurs dépôts, à divers endroits.
- Il se débrouille seul : lorsque vous installez un package, il installe aussi ses dépendances (s'il les trouve).
- Le fichier /etc/apt/sources.list contient les emplacements où sont recherché les packages à installer.



- Chaque ligne a la forme suivante : (deb|deb-src)<URL><distribution><composant>

Gestion de package APT

- APT signifie Advanced Packaging Tool.
- La commande haut-niveau en mode ligne de commande qui est une interface à la commande dpkg.
- Au lieu de spécifier un paquet (local ou distant), il prend en charge des dépôts de packages situés sur un CD, un DVD, dans un répertoire local, sur une source distante sur Internet (ftp, http), etc. L'environnement APT (Advanced Packaging Tool) offre à l'administrateur une solution simple à la gestion des packages.
- Les commandes :
 - apt-get : c'est la commande principale de gestion de package.
 - apt-cdrom : met à jour la liste des packages disponibles stockés sur un CDROM.
 - apt-cache : manipule le cache d'APT.

Les dépôts dans le fichier sources.list :

- **<URL>** : :=protocole ://<hôte>/<chemin> :
 - est le chemin vers la racine du ou des dépôts.
 - Ce peut être une URL de type http ou ftp, mais aussi un chemin local (file), un CD-Rom ou DVD-Rom (CDrom), un chemin ssh, etc.
 - <protocole> : :=cdrom | file | http | ftp | nfs.
- **<distribution>** : La distribution est, comme son nom l'indique, le nom de la distribution Debian, mais il est possible de spécifier d'autres versions de la distribution (=stable | testing | unstable | frozen pour pouvoir récupérer des packages d'autres dépôts, plus récents par exemple. L'architecture peut être précisée. Si elle ne l'est pas, APT se débrouille seul pour rajouter le suffixe nécessaire.
- **<composant>** : Les composants sont les noms des dépôts pour la distribution donnée. (:=main | contrib | non-free | non-US/main | non-US/contrib).

Gestion de package APT

- Mettre à jour un package :

```
root@Nom_machine :~# apt-get update package
```

- Mettre à jour l'ensemble des packages :

```
root@Nom_machine :~# apt-get upgrade
```

- Mettre à jour la distribution :

```
root@Nom_machine :~# apt-get dist-upgrade
```

- Simuler l'installation d'un package :

```
root@Nom_machine :~# apt-get -s install package
```

- Installer (ou mettre à jour) un package :

```
root@Nom_machine :~# apt-get install package
```

Gestion de package APT

- Supprimer un package et ses fichiers de configuration. Les fichiers dépendants seront également supprimés :

```
root@Nom_machine :~# apt-get --purge remove package
```

- Recherches des packages :

```
root@Nom_machine :~# apt-cache search package
```

- Visualiser des informations concernant un package :

```
root@Nom_machine :~# apt-cache show package
```

- Visualiser des informations concernant un package :

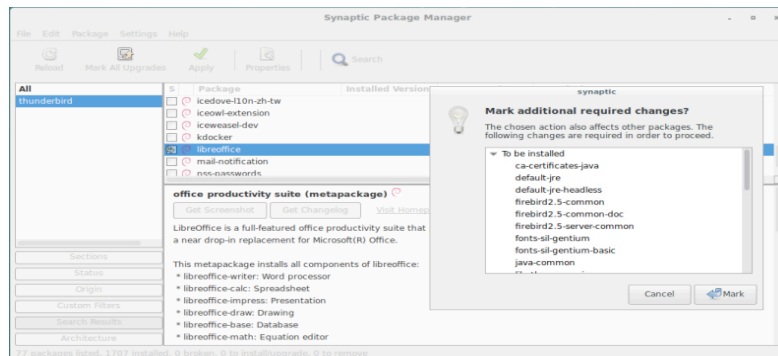
```
root@Nom_machine :~# apt-show-versions package
```

- Visualiser l'emplacement d'un package :

```
root@Nom_machine :~# apt-get --print-uris install package
```

Synaptic Packages Manager APT

- L'outil synaptic est un front-end : une interface graphique qui fait appel aux fonctions de APT.
- Il permet toutes les opérations proposées par APT tout en étant très convivial.



QUESTIONS ?