MODULE 3

Installation de logiciels et gestion de paquets avec apt

Objectifs de ce module :

- ✓ ***Utiliser la commande apt.***
- ✓ ***Faire la gestion des paquets avec apt.***
- ✓ ***Utiliser l'utilitaire graphique pour la gestion des paquets.***

Table des matières

Sujets ***Page***

MODULE 3....1

Installation de logiciels et gestion de paquets avec apt....1

- Introduction....3
- Notion de paquets....3
- Les dépendances....4
- Gestion des paquets en console avec « apt »....5
  - Installer un paquet....5
  - Désinstaller un paquet....7
  - Mise à jour d'un paquet....8
  - Mise à jour de la base de données locale des dépôts....8
  - Recherche de paquet....8
    - Recherche par le nom du paquet....8
    - Recherche par nom du paquet ou de la description du paquet....10
  - Options complémentaires....10
    - L'option show....10
    - L'option contains....11
    - L'option depends....12
    - L'option clean....12

Introduction

La plupart des distributions offrent la possibilité d'installer et de faire la gestion des logiciels par différents utilitaires propres à la distribution utilisées. Par exemple, dans les distributions basées sur RedHat, on utilisera la commande « dnf ». Dans les distributions basées debian, comme Linux Mint, on utilisera la commande « apt ».

Peu importe la distribution utilisée, votre système devrait permettre d'installer, de désinstaller, de mettre à jour des paquets ou de réinstaller des paquets.

Notion de paquets

Nous utilisons le mot paquet depuis le début mais qu'est-ce qu'un paquet ?

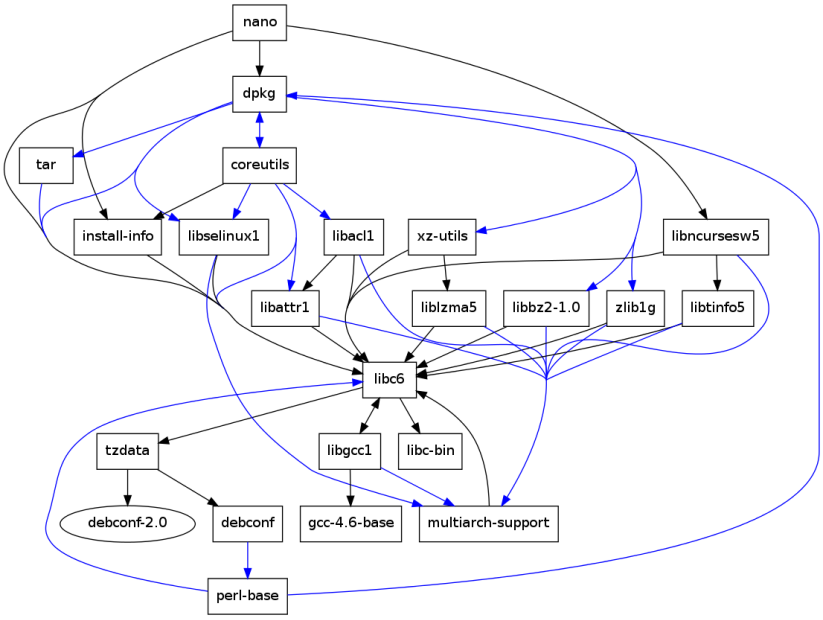
Un paquet est un fichier, habituellement compressé, qui contient les informations et les procédures relatives à l'installation d'un logiciel sur le système.

Un paquet est généralement composé de :

- fichiers exécutables ou sources,
- fichiers de configuration,
- fichiers de documentation,
- dépendances logicielles,
- scripts de pré/post (dé)installation.

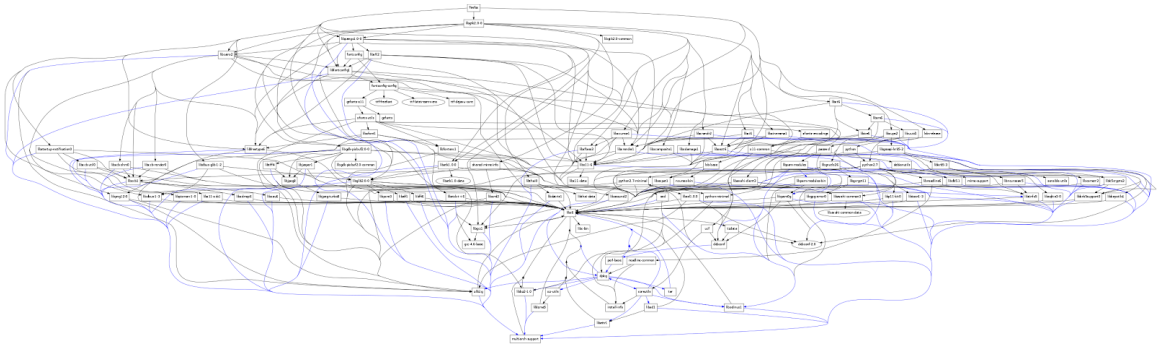
Les dépendances

Les paquets ont souvent besoin de d'autres paquets pour fonctionner. Il faut donc connaitre la liste des paquets dépendants d'un autre. Voici un exemple graphique de dépendance logicielle pour l'éditeur de texte console "nano".



On voit rapidement que l'application « nano » dépend de plusieurs paquets qui dépendent eux-mêmes de plusieurs autres. L'arbre de dépendance peut devenir rapidement un casse-tête inimaginable s'il fallait faire la gestion soit-même de ses paquets !

Voici les dépendances pour l'application « Firefox »

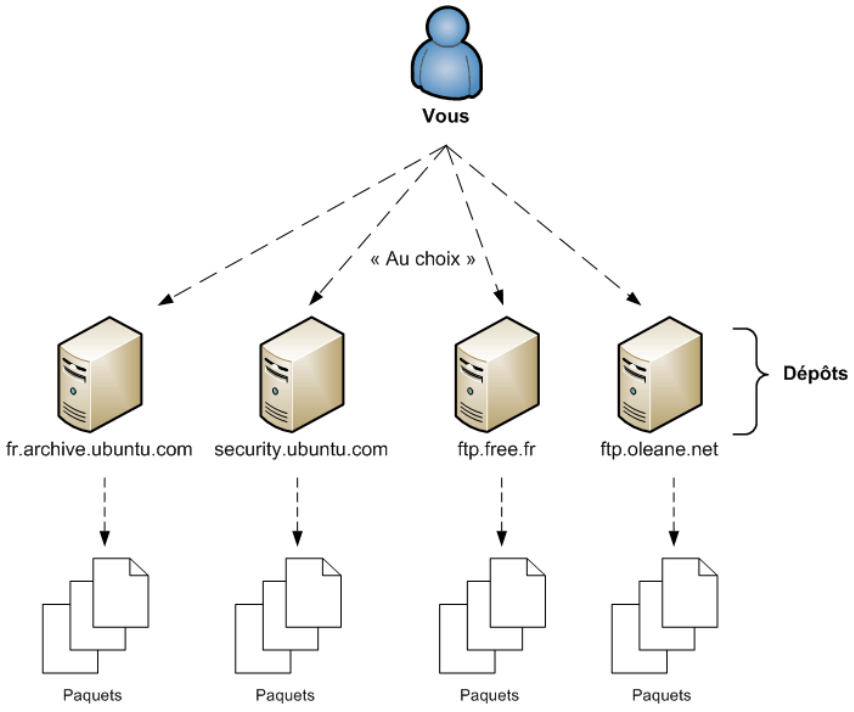


Imaginez installer tout ces paquets manuellement ! Il fallait faire assurément provision de comprimés contre les maux de tête.

Notion de dépôt

Un dépôt est un espace, habituellement un serveur sur Internet, qui contient la liste de tous les paquets de la distribution utilisée. C'est une base de données qui permet de connaître la liste des paquets et leurs dépendances.

On peut se représenter un dépôt un peu comme le schéma montré ci-dessous



Gestion des paquets en console avec « apt »

La commande « apt » permet de faire la gestion de vos logiciels sur votre système. Nous verrons les principales commandes pour faire la gestion de son système.

Installer un paquet

```
apt  install  nom_du_paquet
```

Exemple :

Installer l'application « lbreakout2 »

sudo apt install lbreakout2

```
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances
Lecture des informations d'état... Fait
The following additional packages will be installed:
  lbreakout2-data libmikmod3 libsdl-mixer1.2
Paquets recommandés :
  musescore-soundfont-gm | fluid-soundfont-gm | freepats
Les NOUVEAUX paquets suivants seront installés :
  lbreakout2 lbreakout2-data libmikmod3 libsdl-mixer1.2
0 mis à jour, 4 nouvellement installés, 0 à enlever et 484 non mis à jour.
Il est nécessaire de prendre 2,676 ko dans les archives.
Après cette opération, 5,740 ko d'espace disque supplémentaires seront utilisés.
Souhaitez-vous continuer ? [O/n]
```

Vous obtenez une liste des paquets additionnels qui seront installés (dépendances principales). Il s'agit de répondre « o » pour continuer l'installation. Si on veut répondre oui automatiquement, il s'agit d'ajouter l'option « -y » à la commande d'installation.

```
Dépaquetage de lbreakout2-data (2.6.4-3) ...
Sélection du paquet lbreakout2 précédemment désélectionné.
Préparation du dépaquetage de .../lbreakout2_2.6.4-3_amd64.deb ...
Dépaquetage de lbreakout2 (2.6.4-3) ...
Traitement des actions différées (« triggers ») pour libc-bin (2.23-0ubuntu3) ..
.
Traitement des actions différées (« triggers ») pour gnome-menus (3.13.3-6ubuntu
3) ...
Traitement des actions différées (« triggers ») pour desktop-file-utils (0.22-1u
buntu5) ...
Traitement des actions différées (« triggers ») pour mime-support (3.59ubuntu1)
...
Traitement des actions différées (« triggers ») pour man-db (2.7.5-1) ...
Paramétrage de libmikmod3:amd64 (3.3.8-2) ...
Paramétrage de libsdl-mixer1.2:amd64 (1.2.12-11build1) ...
Paramétrage de lbreakout2-data (2.6.4-3) ...
Paramétrage de lbreakout2 (2.6.4-3) ...
Traitement des actions différées (« triggers ») pour libc-bin (2.23-0ubuntu3) ..
```

Le système télécharge les fichiers, décompresse ces derniers et procède à l'installation et au paramétrage des différents éléments reliés à l'application.

Désinstaller un paquet

Si on veut supprimer l'installation d'une application, il faut alors supprimer le paquet du système.

```
apt  remove  nom_du_paquet
```

Le paquet sera alors désinstallé de votre ordinateur.

Cependant, ceci ne supprimera pas les dépendances devenues inutiles. Pour supprimer le paquet et ses dépendances inutiles :

```
apt  autoremove  nom_du_paquet
```

Même avec autoremove, les fichiers de configuration resteront présents sur le système. Ceci permet de réinstaller l'application et de réutiliser les fichiers de configuration tels qu'ils étaient avant de supprimer l'application. On retrouve donc les fonctionnalités et les informations telles quelles étaient avant la suppression.

Si on veut supprimer aussi les fichiers de configuration, on peut alors taper :

```
apt  purge  nom_du_paquet
```

On peut même jumeler autoremove et purge comme ci-dessous :

```
apt  autoremove  --purge nom_du_paquet
```

Mise à jour d'un paquet

Pour effectuer une mise à jour d'un paquet:

```
apt  upgrade  nom_du_paquet
```

Mise à jour de la base de données locale des dépôts

Attention, il ne faut pas mélanger la mise à jour d'un paquet versus la mise à jour des données locales. La mise à jour des données locales ne fait que mettre à jour la liste des paquets à partir des dépôts sur les serveurs. Il est conseillé de le faire fréquemment.

```
apt  update
```

Recherche de paquet

Il est souvent utile de rechercher un paquet par son nom et de savoir si il est installé ou non et s'il existe un paquet plus récent.

Recherche par le nom du paquet

On peut rechercher un paquet par son nom seulement avec l'option « list ». La recherche se fait alors en cherchant uniquement le nom du paquet qui contient le pattern recherché.

```
apt  list  pattern_recherché
```

Exemple :

Je recherche les paquets qui contiennent « libre »

tapez : sudo apt list libre

IMPORTANT

Il est essentiel de remarquer que c'est le pattern exactement comme vous l'avez écrit qui sera recherché. Dans le cas précédent, c'est le nom libre seulement qui est recherché.

Si on veut être plus large sur la recherche, il faudra alors utiliser les caractères génériques usuels comme « * » et «? ».

Exemple de recherche

Recherchons uniquement les paquets qui contiennent le mot « libre »

```
schasse@bo ~ $ apt list libre
En train de lister... Fait
schasse@bo ~ $
```

Le système répond « En train de lister...Fait ».

On remarque que le système n'a rien trouvé puisque l'on recherchait un ou des paquets qui comporte le nom « libre » exactement sans aucun caractère de plus ou de moins.

Recherchons plutôt les paquets qui commencent par « libre »

```
schasse@bo ~ $ apt list libre*
En train de lister... Fait
libre-dev/xenial 0.4.14-4 amd64
libre-ocaml-dev/xenial 1.2.2-1build1 amd64
libre0/xenial 0.4.14-4 amd64
libre2-1v5/xenial 20160201+dfsg-1 amd64
libre2-dev/xenial 20160201+dfsg-1 amd64
libre2-ocaml/xenial 113.00.00+dfsg-2 amd64
libre2-ocaml-dev/xenial 113.00.00+dfsg-2 amd64
libreact-ocaml/xenial 1.2.0-2build1 amd64
libreact-ocaml-dev/xenial 1.2.0-2build1 amd64
libreactive-streams-java/xenial,xenial 1.0.0-1 all
libreactivedata-ocaml/xenial 0.1-2 amd64
libreactivedata-ocaml-dev/xenial 0.1-2 amd64
libreadline-dev/xenial 6.3-8ubuntu2 amd64
libreadline-gplv2-dev/xenial 5.2+dfsg-3build1 amd64
```

On constate que la liste de paquet trouvé est beaucoup plus longue. Ceci vient du fait que l'on a demandé de rechercher les paquets qui commencent par « libre* ». Autrement dit, tous les paquets qui commencent par « libre » peut importe le reste (l'étoile *).

Recherche par nom du paquet ou de la description du paquet

Ceci permet une recherche plus étendue que la recherche par le nom du paquet seulement comme dans le cas de l'option « list ». La recherche se fait sur non seulement dans le nom du paquet mais aussi dans la description de ce dernier.

Exemple : Recherchons les paquets qui traitent de « jpeg ».

```
schasse@bo ~ $ sudo apt search jpeg
p   gem-plugin-jpeg                      - Environnement graphique pour le multimédia - pri
p   gem-plugin-jpeg:i386                 - Environnement graphique pour le multimédia - pri
p   jpeginfo                             - Imprime des informations et des tests d'intégrit
p   jpeginfo:i386                        - Imprime des informations et des tests d'intégrit
p   jpegjudge                            - Déterminer laquelle de deux images jpegs données
p   jpegjudge:i386                       - Déterminer laquelle de deux images jpegs données
p   jpegoptim                            - Utilitaire pour optimiser des fichiers JPEG
p   jpegoptim:i386                       - Utilitaire pour optimiser des fichiers JPEG
p   jpegpixi                             - Enlever les points chauds des images JPEG avec p
p   jpegpixi:i386                        - Enlever les points chauds des images JPEG avec p
p   libimage-metadata-jpeg-perl          - Perl extension for showing/modifying JPEG (meta)
v   libjpeg-dbg                          -
v   libjpeg-dbg:i386                     -
p   libjpeg-dev                          - Bibliothèque d'exécution JPEG du Independent JPE
p   libjpeg-dev:i386                     - Bibliothèque d'exécution JPEG du Independent JPE
i   libjpeg-progs                        - Programmes de manipulation de fichiers JPEG
p   libjpeg-progs:i386                   - Programmes de manipulation de fichiers JPEG
p   libjpeg-turbo-progs                  - Programmes de manipulation de fichiers JPEG
p   libjpeg-turbo-progs:i386             - Programmes de manipulation de fichiers JPEG
```

Options complémentaires

Quelques options de apt

L'option show

Cette option permet de récupérer un descriptif du paquet qui lui est passé en paramètre, qu'il soit installé sur le système ou non ; elle affiche également d'autres compléments relatifs à ce paquet tels que sa version, son architecture, sa taille, le dépôt le proposant et ses dépendances :

```
apt  show   nom_du_paquet
```

```
schasse@bo ~ $ sudo apt show aurora
Package: aurora
Version: 1.8.8-0ubuntu1
Priority: extra
Section: universe/misc
Origin: Ubuntu
Maintainer: Dustin Kirkland <kirkland@ubuntu.com>
Bugs: https://bugs.launchpad.net/ubuntu/+filebug
Installed-Size: 137 kB
Depends: libc6 (>= 2.7)
Homepage: http://www.curtronics.com/Solar/AuroraData.html
Download-Size: 41.4 kB
APT-Sources: http://archive.ubuntu.com/ubuntu xenial/universe amd64 Packages
Description: communiquer avec un onduleur photovoltaïque (solaire) Aurora Magnet
ek
 Ce programme permettra de communiquer via des connexions série ou USB avec
 un onduleur Aurora photovoltaïque. Il peut récupérer et traiter les
 statistiques recueillies par l'onduleur lui-même.
```

La ligne qui commence par « depends » liste toutes les dépendances du paquet.

L'option contains

```
apt  contains  nom_de_la_commande_ou_de_l'application
```

Si vous vous êtes déjà demandé quel paquet fournit un programme ou une bibliothèque donnée, l'option « contains » devrait répondre à vos attentes : celle-ci interroge en effet les dépôts pour déterminer le paquet proposant la ressource qui lui est passée en paramètre. Une fois le paquet identifié, ne reste plus qu'à l'installer. Notez que dans le cas où l'on souhaite trouver un paquet à partir d'un fichier, cette option attend le chemin complet supposé de ce dernier (ou au moins une expression rationnelle) :

Recherchons le nom des paquets qui contiennent le logiciel « useradd ».

sudo apt contains useradd

```
schasse@bo ~ $ apt contains useradd
bash-completion: /usr/share/bash-completion/completions/luseradd
bash-completion: /usr/share/bash-completion/completions/useradd
passwd: /etc/default/useradd
passwd: /usr/sbin/useradd
passwd: /usr/share/man/de/man8/useradd.8.gz
passwd: /usr/share/man/fr/man8/useradd.8.gz
passwd: /usr/share/man/id/man8/useradd.8.gz
passwd: /usr/share/man/it/man8/useradd.8.gz
passwd: /usr/share/man/ja/man8/useradd.8.gz
passwd: /usr/share/man/man8/useradd.8.gz
passwd: /usr/share/man/ru/man8/useradd.8.gz
passwd: /usr/share/man/tr/man8/useradd.8.gz
passwd: /usr/share/man/zh_CN/man8/useradd.8.gz
passwd: /usr/share/man/zh_TW/man8/useradd.8.gz
```

On remarque qu'il y a 2 paquets qui contiennent la commande useradd. Il s'agit de « bash-completion » et « passwd ». Il s'agira d'aller plus loin dans les informations de ces 2 paquets pour connaître lequel installé selon nos besoins recherchés.

L'option depends

Cette option renvoie, pour un paquet passé en paramètre, la liste de ses dépendances :

Exemple : Recherchons les dépendances du paquet nommé « nano ».

```
schasse@bo ~ $ apt depends nano
nano
  Dépend: libc6
  Dépend: libncursesw5
  Dépend: libtinfo5
  Est en conflit avec: <pico>
  Suggère: spell
  Remplace: <pico>
```

On remarque donc que nano dépend de 3 paquets : « libc6 », « libncursesw5 » et « libtinfo5 ».

L'option clean

APT conserve les différentes informations relatives aux dépôts, et en particulier les paquets et les composants des paquets, dans des fichiers de cache durant un certain temps (dans le dossier /var/cache/apt/). De cette façon, APT n'a pas à interroger systématiquement les dépôts en ligne à chaque utilisation, ce qui rallongerait considérablement ses traitements.

Il peut néanmoins s'avérer nécessaire de purger ce cache manuellement, pour forcer APT à récupérer de nouvelles données dépôts à jour.

On peut taper la commande suivante :

```
apt clean all
```

Ainsi, au prochain lancement de APT, les informations dépôts seront de nouveau générées.