

Administration Système sous Linux (Ubuntu Server)

Grégory Morel

2021-2022

CPE Lyon - 3IRC / 4ETI / 3ICS

Cours 3

Gestion des paquets

Gestion des paquets

Préliminaire : exécuter des commandes en mode administrateur

La commande **sudo** (pour *Substitute User and DO*) permet d'exécuter une commande en prenant l'identité d'un autre utilisateur :

```
sudo commande [utilisateur]
```

En particulier, **utilisateur** est par défaut le compte administrateur du système, **root**¹. Cette commande permet donc d'exécuter une commande avec les privilèges de l'administrateur, **à condition d'en avoir la permission** !

Elle est indispensable pour pouvoir installer des paquets, par exemple :

```
sudo apt install vlc
```

1. Voir la section **Gestion des utilisateurs**

Installation d'un logiciel sous Linux

Dans les années 90 (ou pour les puristes) \Leftrightarrow compiler le code source

Avantages

- on reste indépendant de la distribution Linux
- on peut adapter la compilation à la configuration matérielle (par ex. utiliser des fonctionnalités avancées du processeur)
- on peut disposer de la toute dernière version d'un logiciel

Inconvénients

- surplus de travail et de temps lié à la compilation
- pas de mise à jour (il faut recommencer le processus pour chaque nouvelle version)
- gestion des dépendances plus délicate
- pas de désinstallation automatique

Il serait plus simple d'avoir un système automatisé !

Notion de paquet

Un **paquet** est un fichier contenant tout le nécessaire à l'installation *cohérente* d'un logiciel.

Un paquet Debian est simplement une **archive au format ar**, et contient trois fichiers :

```
$ ar -t scilab_5.5.2-4_all.deb
debian-binary          #version du package
control.tar.gz         #sommes de contrôle
data.tar.xz            #le contenu du paquet lui-même
```

A noter

Les paquets ont une extension **.deb**

Sur <https://packages.ubuntu.com/> on trouve les paquets pour chaque version d'Ubuntu

dpkg (Debian Package Manager)

Base du gestionnaire de paquets des systèmes Debian (donc Ubuntu).

Permet de créer, installer, gérer et supprimer les paquets.

<code>dpkg -i mypackage.deb</code>	installe ou met à jour le paquet <i>mypackage</i>
<code>dpkg -R dossier</code>	installe les packages situés dans <i>dossier</i> et ses sous-dossiers
<code>dpkg -r mypackage.deb</code>	supprime le paquet <i>mypackage</i> mais conserve les fichiers de configuration
<code>dpkg -P mypackage.deb</code>	purge le paquet <i>mypackage</i> (i.e. supprime aussi les fichiers de configuration)

dpkg

dpkg est un outil de **bas niveau** (comparé à APT). En particulier, **dpkg ne gère pas les dépendances** : il faut les résoudre à la main !

dpkg s'utilise surtout en cas de conflit entre des paquets, en agissant sur un seul paquet sans bouleverser les dépendances.

Utile!

la commande **dpkg-reconfigure** permet de... reconfigurer un paquet

A noter

Les fichiers de configuration de **dpkg** sont dans **/var/lib/dpkg**

Les fichiers d'historique (**dpkg.log**) se trouvent dans **/var/log**

Quelques commandes dpkg

dpkg permet aussi d'avoir des informations détaillées sur les paquets :

<code>dpkg -l</code>	affiche les paquets installés
<code>dpkg -l pattern</code>	affiche les paquets (installés ou non) dont le nom correspond à <i>pattern</i>
<code>dpkg -S commande</code>	permet de retrouver le paquet qui a installé <i>commande</i>
<code>dpkg -c package.deb</code>	liste le contenu de <i>package.deb</i>
<code>dpkg -L package-name</code>	liste les fichiers qui ont été installés par <i>package-name</i>
<code>dpkg-deb -x package.deb</code>	extraie l'arborescence du paquet ¹
<code>dpkg-deb -e package.deb</code>	extraie les fichiers de contrôle

1. Extraire un paquet dans le dossier racine / ne conduit **PAS** à une installation correcte

Problème de dpkg

Comme on l'a vu, dpkg ne gère pas les dépendances !

En 1999, Debian 2.1 a introduit APT (Advanced Package Tool) pour simplifier la gestion des paquets, la gestion des dépendances, la gestion des dépôts...

apt vs apt-get

Les commandes originelles APT sont nombreuses (**apt-get**, **apt-cache**...). Ubuntu 14.04 a introduit une nouvelle et unique commande, **apt**, plus simple (mais moins puissante).

Commande apt	Commande remplacée	Description
<code>apt install</code> ¹	<code>apt-get install</code>	Installe un paquet ²
<code>apt remove</code>	<code>apt-get remove</code>	Supprime un paquet
<code>apt purge</code>	<code>apt-get purge</code>	Supprime un paquet et sa config.
<code>apt update</code>	<code>apt-get update</code>	Met à jour l'index des dépôts ³
<code>apt upgrade</code>	<code>apt-get upgrade</code>	Met à jour les paquets qui le peuvent
<code>apt autoremove</code>	<code>apt-get autoremove</code>	Supprime les paquets non désirés
<code>apt full-upgrade</code>	<code>apt-get dist-upgrade</code>	Met à jour la distribution
<code>apt search</code>	<code>apt-cache search</code>	Recherche textuelle
<code>apt show</code>	<code>apt-cache show</code>	Affiche les détails d'un paquet
<code>apt list</code>		Liste les paquets sur critères
<code>apt edit-sources</code>		Edite puis vérifie sources.list

1. L'option `-y` permet de répondre automatiquement 'oui' aux questions

2. Le journal des modifications est le fichier `/var/log/apt/history.log`

3. Les index sont générés et stockés dans les fichiers `/var/lib/apt/*_Packages`

Dépôts

Les dépôts sont indiqués dans le fichier `/etc/apt/sources.list`¹

Format des informations de dépôts

Les dépôts renseignés dans `sources.list` ont tous le même format :

`deb`² *URL nom_distribution composant1 composant2...*

`nom_distribution` devrait toujours être le même et correspondre à la distribution utilisée

Composants	Maintenus par Canonical	Maintenus par la communauté
Libres	main	universe
Non-libres	restricted	multiverse

1. Les paquets non officiels, comme Google Chrome, créent généralement chacun un fichier supplémentaire dans `/etc/apt/sources.list.d`
2. Ou `deb-src` pour des dépôts de code source

Problème

Les dépôts contiennent rarement la *dernière* version d'un logiciel (il faut que le logiciel ait été déclaré 100 % compatible avec la distribution)

⇒ impossible pour l'utilisateur de bénéficier des derniers correctifs de sécurité et fonctionnalités

2 solutions possibles :

- compiler et installer la dernière version manuellement à partir des sources (et les dépendances...!)
- PPA = **Personal Package Archives**

Un PPA est un dépôt hébergé sur la plateforme [Launchpad.net](https://launchpad.net)¹

N'importe qui peut créer un projet sur Launchpad. Il faut veiller à n'installer que des paquets dignes de confiance!

L'ajout d'un PPA se fait par la commande :

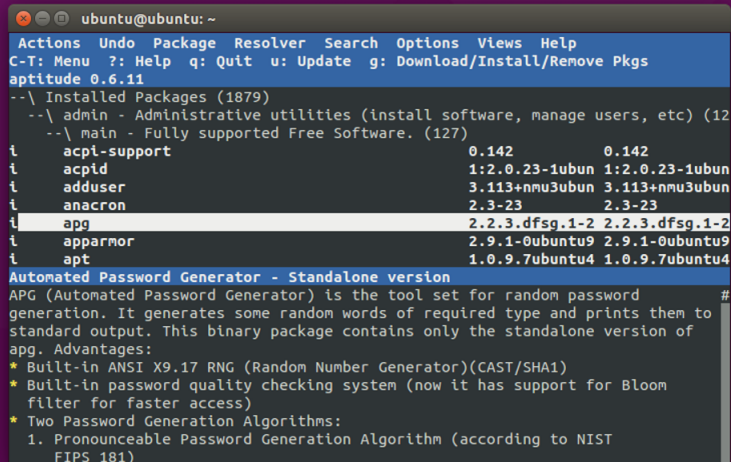
```
sudo add-apt-repository -u ppa:<nom_du_dépôt>
```

A retenir

L'option `-u` permet d'actualiser la liste des paquets après l'ajout du PPA (on peut le faire a posteriori avec `apt update`)

1. La liste des PPA est accessible ici : <https://launchpad.net/ubuntu/+ppas>

Gestionnaire de paquets offrant une interface en ligne de commande (semblable à **apt**) et une interface graphique :



```
ubuntu@ubuntu: ~  
Actions Undo Package Resolver Search Options Views Help  
C-T: Menu ?: Help q: Quit u: Update g: Download/Install/Remove Pkgs  
aptitude 0.6.11  
--\ Installed Packages (1879)  
  --\ admin - Administrative utilities (install software, manage users, etc) (12)  
    --\ main - Fully supported Free Software. (127)  
i   acpi-support                0.142                0.142  
i   acpid                      1:2.0.23-1ubun      1:2.0.23-1ubun  
i   adduser                    3.113+nmu3ubun      3.113+nmu3ubun  
i   anacron                    2.3-23              2.3-23  
i   apg                        2.2.3.dfsg.1-2      2.2.3.dfsg.1-2  
i   apparmor                   2.9.1-0ubuntu9      2.9.1-0ubuntu9  
i   apt                        1.0.9.7ubuntu4       1.0.9.7ubuntu4  
  
Automated Password Generator - Standalone version  
APG (Automated Password Generator) is the tool set for random password  
generation. It generates some random words of required type and prints them to  
standard output. This binary package contains only the standalone version of  
apg. Advantages:  
* Built-in ANSI X9.17 RNG (Random Number Generator)(CAST/SHA1)  
* Built-in password quality checking system (now it has support for Bloom  
  filter for faster access)  
* Two Password Generation Algorithms:  
  1. Pronounceable Password Generation Algorithm (according to NIST  
    FIPS 181)
```

Quelques commandes de base :

/ : recherche

n : poursuivre la recherche (**n**ext)

+ : sélectionner pour installation

g : une fois pour l'aperçu des changements, deux fois pour confirmer

- : suppression simple (= **remove**)

_ : suppression complète (= **purge**)

u : mise à jour de la liste des paquets (= **update**)

U : mise à jour des paquets (= **upgrade**)

Indicateur	Description
i	Paquet installé et dépendances satisfaites
c	Paquet supprimé sauf fichiers de configuration
p	Paquet et fichiers de configuration supprimés
v	Paquet virtuel
B	Paquet avec dépendances cassées
A	Paquet installé automatiquement

Un **paquet virtuel** est un nom générique pour un ensemble de paquets qui fournissent la même fonctionnalité. Par exemple, **firefox**, **chromium** ou **lynx** font partie du paquet **www-browser**.

Backports

Problème

Chaque release d'Ubuntu est figée avec son contenu logiciel : les nouvelles versions majeures des logiciels (dont les correctifs de sécurité et les nouvelles fonctionnalités) ne seront pas disponibles pour cette release !

La communauté Ubuntu met à disposition des **Backports**, i.e. des portages de nouvelles versions de logiciels pour des anciennes releases d'Ubuntu

Attention

Les Backports sont distribués **sans garantie de sécurité**. Par ailleurs, bien qu'ils soient testés par la communauté, il y a toujours des **risques d'incompatibilité** avec d'anciens logiciels

Pour activer les Backports, il faut que le fichier **sources.list** contienne une ligne :

```
deb ... nom_distribution-backports...
```

Aparté : et chez Red Hat ?

Debian	Red Hat
dpkg	rpm
apt	yum (ou dnf à partir de Fedora 22)

Bon à savoir

Il existe un paquet, **alien**, permettant de convertir un paquet Red Hat, Slackware, etc. en un paquet Debian

Format de paquet développé par Canonical

Objectif : alternative sécurisée aux PPA \Rightarrow les applications s'exécutent dans une *sandbox*, isolée du système

💡 Un snap peut contenir les *dépendances* du logiciel :

- + plus besoin de gérer les dépendances
- le paquet est plus lourd
- les dépendances sont dupliquées
- + on peut faire cohabiter plusieurs versions sur le même système

💡 Se rapproche davantage de la philosophie Windows...