

## 1 Introduction

L'objectif de ce TP est de vous apprendre à installer des paquets dans un système d'exploitation Linux.

Un paquet est une archive comprenant des fichiers informatiques, les informations et procédures nécessaires à l'installation d'un logiciel sur un système d'exploitation, en s'assurant de la cohérence fonctionnelle du système ainsi modifié.

Un gestionnaire de paquets est un (ou plusieurs) outil(s) automatisant le processus d'installation, désinstallation, mise à jour de logiciels installés sur un système informatique.

Ce gestionnaire de paquets permet d'effectuer différentes opérations sur les paquets disponibles comme l'installation, la mise à jour et la désinstallation. Il permet d'utiliser des paquets provenant de différents supports (CD ou DVD d'installation, dépôts sur internet, partage réseau ...). Il vérifie l'intégrité des applications à installer et des dépendances logicielles.

On trouve deux grands types de système de paquets :

- **rpm** (redhat package manager) : Redhat Enterprise Linux, Fedora, Centos, ...
- **deb** (debian) : Debian, Ubuntu, Mint, ...

Les gestionnaires de paquets associés :

- **dpkg** est utilisé pour installer, supprimer et fournir des informations à propos des paquets \*.deb qui sont supportés par les distributions basées Debian. C'est un outil de bas niveau et on préfère utiliser des outils plus avancés comme apt (apt-get, apt-cache, etc.) ou dérivés graphiques (aptitude, synaptic ...).

Remarque : Pour reconfigurer un paquet installé, il arrive qu'on soit obligé d'utiliser la commande **dpkg-reconfigure**.

- **rpm** permet d'installer, mettre à jour, désinstaller, vérifier et rechercher des paquets, avec les droits de l'utilisateur root.

Un gestionnaire de paquet avancé comme apt (Debian et dérivés) ou yum ou dnf (Red Hat et dérivés) gère des sources de logiciels (la plupart du temps déjà compilés) et leur authenticité. Le lieu où sont placés ces sources est appelé *dépôt de paquets*. Cette source est la plupart du

temps une source locale comme un CD ou un DVD, un serveur Internet HTTP/FTP ou encore un miroir de dépôt local. Certains concepteurs de logiciels fabriquent eux-mêmes les binaires d'installation pour les distributions et hébergent leurs propres dépôts de paquets (appelés PPA - *Personal Package Archive*).

`apt` simplifie l'installation, la mise à jour et la désinstallation de logiciels en automatisant la récupération de paquets à partir de sources (sur Internet, le réseau local, des CD-ROM, etc.), la gestion des dépendances et parfois la compilation. Il est recommandé officiellement par le projet Debian. Les sources à partir desquelles `apt` va chercher les paquets sont définies dans le fichier `/etc/apt/sources.list`. Ubuntu maintient officiellement les paquets `main` (logiciels libres) et `restricted` (logiciels non-libres) et la communauté Ubuntu fournit les paquets `universe` (libres) et `multiverse` (non-libres).

Plus d'informations détaillées sur :

<https://linux.goffinet.org/administration/installation-logiciels/paquets-linux/>

Finalement, `snap` et `flatpak` sont des systèmes d'emballage et de déploiement de logiciels. Vous avez plus d'informations sur le lien suivant :

<https://www.nextinpact.com/article/30327/109059-paquets-appimage-snap-et-flatpak-quels-avantages-inconvenients-et-differences>

## 2 Installation en utilisant un gestionnaire de paquets

### 2.1 Introduction

Nous allons nous concentrer sur le gestionnaire des paquets d'Ubuntu pour profiter de l'installation faite sur une machine virtuelle dans le TP précédent ainsi que de la distribution Ubuntu existante sur vos portables CY Tech.

Un paquet est installé sur Ubuntu en utilisant les commandes :

- `"apt-get install"` (ancienne syntaxe) ou `"apt install"` (nouvelle syntaxe), précédées de `sudo` si nous ne sommes pas les administrateurs de la machine.

Pour désinstaller un paquet, les mêmes commandes sont utilisées mais avec le mot clef `"remove"`.

Prenez en compte que, avant d'installer un paquet, c'est une bonne habitude de synchroniser la liste des paquets existants dans notre ordinateur avec celle du fournisseur des dépôts des paquets, afin de s'assurer de mettre la liste des logiciels à jour ainsi que les liens de téléchargement. Il faudra donc exécuter la commande suivante avant d'exécuter `"apt install"` :

- `"apt-get update"` (ancienne syntaxe) ou `"apt update"` (nouvelle syntaxe), toujours précédées de `sudo` si nous ne sommes pas administrateurs de la machine.

Pour installer un paquet (donc un logiciel, librairie, ou autre), il faut connaître son nom. Pour chercher un paquet il faut utiliser les commandes :

- `"apt-cache search"` ou `"apt search"`.

Notez que `apt` cherche les dépôts dans les sources énumérées dans `/etc/apt/sources.list`. Regardez ce fichier et vérifiez son contenu.

Finalement, pour lister tous les paquets installés dans votre système, vous pouvez utiliser la commande :

- `"apt list --installed"`.

Sauf si c'est indiqué autrement, toutes les démarches désormais proposées pourront être faites soit sur votre portable CY-Tech, soit (pour plus de sécurité) sur la machine virtuelle contenant votre installation d'Ubuntu.

## 2.2 À faire

Chercher et installer le paquet `glances` ainsi que son paquet de documentation. Par défaut, la version installée sera la plus récente. Rappelez-vous de précéder les commandes avec `"sudo"` si votre utilisateur n'est pas un administrateur de la machine.

Si nous souhaitons installer une version en particulier, les versions disponibles peuvent être recherchées avec la commande `"apt-cache policy nom_du_paquet"`. Pour installer une version en particulier nous utiliserons `"apt install nom_du_paquet=version_du_paquet"`. Comme mise en pratique, installer le paquet `"apache2"` mais la version la plus ancienne disponible.

Une fois `"apache2"` installée, vous pouvez le supprimer de votre système avec la commande `"apt remove nom_du_paquet"` en laissant les fichiers de configuration dans le système ou `"apt purge nom_du_paquet"` si vous voulez vous assurer que les fichiers de configuration de ce paquet soient aussi supprimés.

# 3 Installation à partir de paquets non présents dans les dépôts officiels

## 3.1 Dépôts personnels

Il arrive parfois que le logiciel que nous souhaitons installer ne soit pas disponible dans les dépôts officiels ou que la version disponible soit très ancienne.

Les dépôts personnels de paquets logiciels, ou *Personal Package Archives* (abrévés PPA) sont un service qui prend le code source déposé par les développeurs de logiciels et génère des paquets `.deb` que les utilisateurs d'Ubuntu pourront installer à travers leur gestionnaire de paquets logiciels. Les avantages des PPA est qu'ils sont des dépôts de paquets logiciels dont le contenu est très récent. Contrairement à un paquet isolé (à voir plus tard dans ce TP), si un paquet installé à partir d'un PPA subit une mise à jour, vous en bénéficierez automatiquement. En revanche, les applications provenant des PPAs ne bénéficient pas de la validation officielle Ubuntu. Prenez en compte qu'installer une application hors des dépôts officiels peut présenter des risques d'instabilité de votre système.

Pour ajouter un PPA à vos sources logicielles, il faudra saisir la commande suivante :

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

suivi de `"sudo apt update"`.

Comme exemple, nous allons installer la dernière version disponible d'un logiciel gestionnaire de passwords, `keepassxc`, à partir de son PPA. D'abord, vérifiez avec `"apt search"` la version de `keepassxc` disponible sur les dépôts officiels d'Ubuntu. Ensuite, allez sur <https://keepassxc.org/download/#linux> et, sous la rubrique *Ubuntu*, cliquez sur le lien *"Official Ubuntu PPA"*. Vous serez redirigés vers une page de *launchpad.net*. Suivez les instructions sous la rubrique *"Adding this PPA to your system"* pour ajouter le PPA à votre système (alternativement, notez que la rubrique *"Technical details about this PPA"* vous explique comment ajouter le dépôt directement dans votre fichier `sources.list`). Finalement, installez `keepassxc` et vérifiez que la version est plus moderne que celle des dépôts officiels d'Ubuntu.

Pour lister les dépôts PPA présents sur votre système, dans un terminal, saisir la commande suivante :

- `ls -l /etc/apt/sources.list.d/`

Vérifiez que le PPA de `keepassxc` est effectivement là. Finalement, supprimez `keepassxc` et son PPA. Cherchez sur la page du manuel de `add-apt-repository` comment supprimer un PPA.

### 3.2 Téléchargement direct d'un paquet

Occasionnellement, le logiciel que l'on souhaite installer n'est ni disponible sous les dépôts officielles ni sous un PPA, mais téléchargeable au format `.deb` (pour les architectures de type `debian`) ou `.rpm` (pour les architecture de type `Red Hat`). Ces formats sont des formats de paquets cohérents avec votre système.

Nous allons installer le logiciel haroopad. Ce logiciel ne fait partie d'aucun dépôt officiel, il faut donc se rendre sur le site d'haroopad<sup>1</sup> et télécharger l'archive correspondante pour votre système. Pour ce faire, il faut donc connaître votre système précisément. Normalement la commande `uname -a`, vous donne les informations nécessaires. Pour vous, vous allez télécharger le fichier *Linux Debian Package (64 bits)*. Une fois ce fichier téléchargé, il suffit de l'installer avec la commande `dpkg -i nomDuPaquet`.

## 4 Installation à partir de code source

### 4.1 Récupération des codes sources

L'installation à partir de code source est un étape délicate, qui requière parfois une expertise. Il existe plusieurs façons de récupérer des codes sources :

- à partir d'un site web qui met à disposition les sources du logiciel
- à partir d'un gestionnaire de version tel que `git`

Dans tous les cas, vous allez récupérer les codes sources dans une archive (généralement en `tgz` ou `zip`). La première étape est de décompresser l'archive. Ensuite, il n'y a pas de méthodologie standard, tout dépend de ce que vous avez récupéré... La lecture du `README` est alors indispensable.

Allez à l'url `https://github.com/todotxt/todo.txt-cli`, clonez<sup>2</sup> ou téléchargez le projet. Puis suivez les instructions contenues sur le fichier `README.md`

### 4.2 Mise à jour du système

Dans cette section nous allons faire une mise à jour **de votre machine virtuelle Ubuntu** (ne le faites pas sur vos machines CY Tech!).

#### 4.2.1 Mise à jour d'Ubuntu

Avant de faire la mise à jour, vous pouvez vérifier la version d'Ubuntu installée sur votre machine virtuelle avec les commandes `lsb_release -a` ou `cat /etc/lsb-release` ou `cat /etc/issue`. Nous allons faire un *upgrade* de notre version 16.04 d'Ubuntu (*Xenial*) à la version 18.04 (*Bionic Beaver*). Pour cela, il faudra s'assurer que tous les paquets de la version actuelle

---

1. `http://pad.haroopress.com/user.html`  
2. `git clone URL`

sont à jour. Pour ce faire, nous allons exécuter `sudo apt-get update` pour resynchroniser la liste de paquets. Ensuite, on utilisera la commande `sudo apt dist-upgrade` pour installer les versions les plus récentes de tous les paquets en prenant en compte les changements de dépendances dans les nouvelles versions des paquets.

Une fois que la version 16.04 soit à jour, assurez vous que vous avez le paquet `update-manager-core` installé : `sudo apt install update-manager-core`.

Ensuite, utiliser un éditeur de texte pour modifier le fichier de configuration suivant : `sudo vim /etc/update-manager/release-upgrades`. À la fin de ce fichier, changer la valeur de la variable `Prompt` de `"lts"` à `"normal"`. Sauvegardez et fermez le fichier, et exécutez la commande suivante pour démarrer le *upgrade* à la version 18.04 : `do-release-upgrade`.

Une fois la mise à jour finie, vérifiez que vous avez la nouvelle version installée avec `lsb_release -a` et regardez le contenu de l'archive `/etc/apt/sources.list`, qui devra aussi avoir changé.