Objectif :

Savoir à quoi sert les paquets, dépôts, sources.lists, et comment les utiliser.

Consignes :

Ce document est un cours vous expliquant le fonctionnement des commandes. Afin de mieux le comprendre, mettez en application chaque commande sur votre machine, pour que vous puissiez en voir les effets.

À vous de faire :

Lors de l’installation, vous avez configuré un ‘miroir’, c’est un nom qu’on donne au dépôt, sur lequel sont déposés les logiciels, les mises à jour …

Et pour installer tout ça sur votre machine, vous utilisez la commande apt, qui demande à votre machine d’aller chercher le logiciel en question sur le miroir.

Donc, vous savez utiliser apt et le miroir. Rentrons un peu plus dans les détails pour comprendre le fonctionnement.

# I - Paquets

Sous Windows, pour installer un logiciel, on va sur internet chercher le logiciel en question, on télécharge son programme d’installation en .exe, qui lui va se charger d’installer le logiciel avec les paramètres qu’on lui donne.

Sous Linux, c’est quasiment pareil. À peu de chose près. Ce n’est pas vous qui allez chercher sur internet pour trouver le logiciel, mais votre machine elle-même. On y reviendra. Elle cherche donc le logiciel demandé, elle trouve son ‘paquet’ correspondant (son programme d’installation). Ce paquet, est un dossier compressé en .deb, pour DEBian. Car ce fonctionnement pour le téléchargement et l’installation des logiciels, est valable sur les distributions Debian. Mais chaque distribution à son propre fonctionnement pour cela. C’est l’une des différences principales entre les distributions.

Maintenant, nous allons parler de paquets, et non pas de logiciels ou de programme, et de dépôt ou miroir.

Quand vous recherchez un paquet, il vous en propose d’en installer d’autres, vous avez pu le voir notamment pour openssh-server. Il fait cela, car les paquets ont des dépendances. C’est-à-dire qu’ils dépendent les un des autres, ou bien d’une bibliothèque de programmes. Hormis le moment où vous faites la recherche de votre paquet, et que votre machine vous répond qu’il là trouvé ainsi que x autres paquets, vous ne voyez pas ces dépendances. C’est en effet à ce moment-là, que les dépendances sont visibles, le paquet que vous souhaitez installer, indique qu’il dépend de x autres, et donc le dépôt vous propose d’installer aussi ces x autres paquets, pour que tout puisse fonctionner. Il vous demande simplement l’autorisation d’installer ces x autres paquets, et après vous n’en entendez plus parler.

# II - Dépôts

Revenons au dépôt, il s’agit donc d’un endroit où tous les paquets sont rassemblés, et sont accessibles gratuitement, car il s’agit de logiciel libre et gratuit. Il existe énormément de dépôts différents, car il faut permettre à tout le monde de pouvoir accéder aux dépôts. Chaque personne peut créer son dépôt, il suffit de dédier un serveur à cela et de le rendre accessible sur internet. C’est pour cela que lors de l’installation de votre machine, vous avez pu voir par exemple des dépôts d’université.

Étant donné que chaque personne peut avoir son dépôt, chaque dépôt à des paquets différents, quand bien même en majorité, les dépôts sont des copies les uns des autres.

Cependant, dans de très rares occasions, vous pouvez ne pas trouver sur votre dépôt le paquet qui vous intéresse. Dans ce cas-là, vous allez le chercher sur internet et le télécharger en .deb, ensuite vous lancer l’installation manuellement sur votre machine.

# III - Sources.list

Comme on l’a vu, on configure le dépôt lors de l’installation de la machine. Mais ce n’est pas un choix figé, on peut le modifier grâce au fichier /etc/apt/sources.list.

Vous le voyez sur votre machine, chaque ligne commence par deb ou deb-src.

Deb est le lien pour télécharger la version compilée des paquets, c’est la version prête à l’emploi.

Deb-src lui permet de récupérer le code source du programme.

Donc on a le type de téléchargement qu’on souhaite faire. Ensuite, l’adresse de notre dépôt. Vient ensuite la distribution qu’on utilise, pour nous c’est Buster. Si on utilise buster-updates c’est pour avoir accès aux mises à jour de sécurité disponibles entre les versions intermédiaires. Et après, il s’agit d’une indication de votre machine pour savoir dans quelle section du dépôt chercher, ici, la section principale.

# IV - apt

Vous connaissez apt pour l’avoir utilisé pour installer des paquets.

#### Comment fonctionne-t-il ?

Lors de l’installation de votre machine, et de la configuration de votre dépôt, la commande apt a téléchargé la liste des paquets disponibles sur le dépôt que vous avez choisi. Elle a mis cette liste dans son cache.

Afin d’être toujours à jour, et d’être sûr de trouver la dernière version de votre paquet, ses caractéristiques, ses dépendances, il est nécessaire de temps en temps de mettre à jour votre cache, en faisant :

apt update

Une fois votre cache mis à jour, vous pouvez installer le paquet désiré. Vous êtes sûr d’avoir la dernière version possible. Pour cela vous le savez, vous utilisez la commande :

apt install [nom\_paquet].

Si vous ne connaissez pas exactement le nom de votre paquet (ils peuvent être tordus), vous utiliserez la commande :

apt-cache search [nom\_de\_votre\_paquet]

Il vous sortira une liste de paquet, ainsi qu’une description les concernant.

Lorsque vous installez un logiciel avec apt install, cela suit plusieurs étapes :

* Vous demandez le paquet que vous souhaitez
* Apt vérifie dans son cache si le paquet existe.
* Il vérifie aussi s’il y a des dépendances
* Si oui, il vous en informe et vous demande confirmation pour leurs installations
* Apt télécharge les paquets depuis le dépôt et toutes ses dépendances
* Il décompresse les paquets, les installes et les paramètres
* Et voilà il a fini, il a installé votre paquet, et celui-ci est prêt à fonctionner. Et la seule chose qu’il vous a demandée c’est la confirmation. Il a tout fait tout seul !

Pour désinstaller un paquet, c’est toujours avec ‘apt’ et là il est suivi de ‘remove’.

apt remove [nom\_paquet]

Cela désinstallera le paquet en question.

Pour désinstaller les dépendances qui ne servent plus à rien si le paquet de base n’est plus là, vous utiliserez :

apt autoremove [nom\_paquet]

Cela supprimera de façon plus propre votre paquet, et libéra plus d’espace, car vous n’aurez pas de dépendances inutiles.

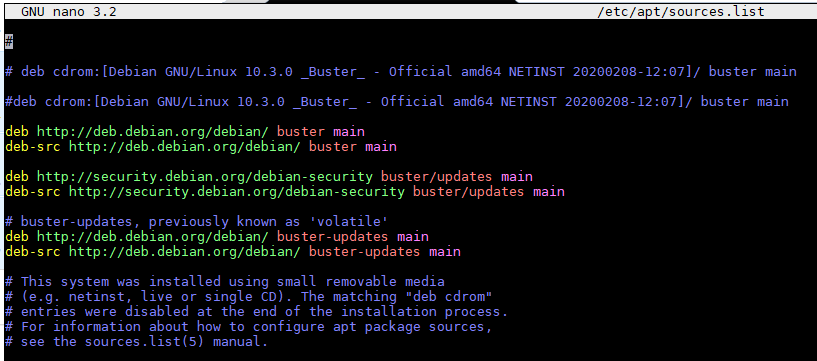
Pour mettre à jour vos paquets, c’est toujours avec la même commande, suivis de upgrade.

apt upgrade

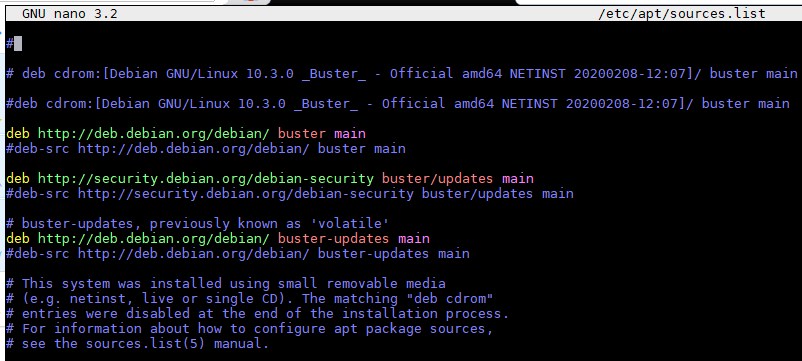
Tous les paquets installés sur votre machine seront mis à jour grâce à cela.

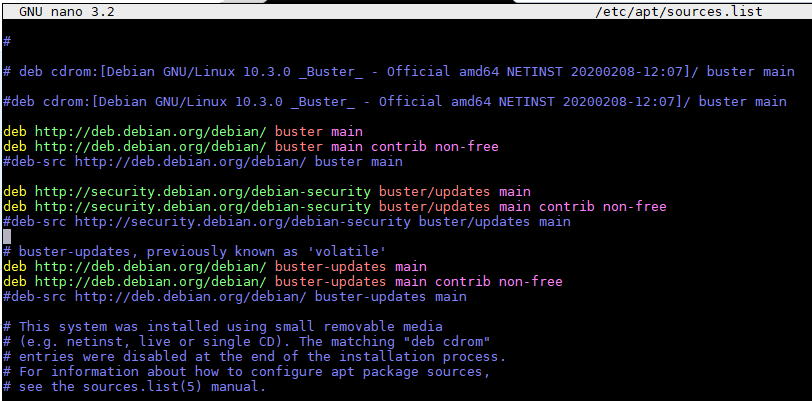
Pour être logique, avant de lancer un upgrade, pensez à mettre le cache à jour en faisant un update !

# V - Pour aller plus loin

Par défaut le fichier ressemble à celui-ci dessous, car lors de l’installation nous avons choisi de configurer le miroir deb.debian.org : Ici le fichier est configuré pour la version 10 de debian, nom de code buster, la version 11 remplacer le nom de code par bullseye et pour la version 12 par bookworm.

Lors du cours sur la configuration d’une machine Debian, il vous a été demandé de simplifier ce fichier en commentant grâce à un # les lignes commençant par deb-src, pour ne plus avoir les paquets au format binaire de téléchargés. Votre fichier correspond donc actuellement à ça :



Mail il arrive qu’il faille ajouter les autres dépôts optionnels pour obtenir plus de paquets parce que vous ne trouvez pas celui dont vous avez besoin, pour cela, il faut ajouter **contrib** et si besoin **non-free** comme l’exemple ci-dessous :

Bon à savoir :

* La section **main** comprend l'ensemble des paquets qui se conforment aux DFSG - Directives Debian pour le logiciel libre et qui n'ont pas besoin de programmes en dehors de ce périmètre pour fonctionner. Ce sont les seuls paquets considérés comme faisant partie de la distribution Debian.
* La section **contrib** comprend l'ensemble des paquets qui se conforment aux DFSG, mais qui ont des dépendances en dehors de main (qui peuvent être empaquetées pour Debian dans non-free).
* La section **non-free** contient des logiciels qui ne se conforment pas aux DFSG