

Accueil > Cours > Reprenez le contrôle à l'aide de Linux ! > Installer des programmes avec apt-get

Reprenez le contrôle à l'aide de Linux !

🕒 30 heures 📊 Facile

Mis à jour le 29/06/2021



Le contenu de ce cours n'est plus à jour

Nous avons archivé ce cours et n'actualiserons plus son contenu.

Accédez au contenu le plus récent en découvrant ce cours :



SYSTÈMES & RÉSEAUX

Initiez-vous à Linux

📊 Easy 🕒 8 heures

Dans ce cours débutant, découvrez Linux : un système d'exploitation gratuit et fascinant qui vous donnera un contrôle sans précédent sur votre ordinateur ! Créé par des passionnés d'informatique, Linux est un vecteur important de la philosophie du libre et l'alternative parfaite à Windows ou macOS.

[VOIR LE NOUVEAU COURS](#)

Installer des programmes avec apt-get

Quand vous êtes sous Windows et que vous voulez télécharger un nouveau programme, que faites-vous ?

En général une petite recherche sur un moteur de recherche, un tour sur les sites de téléchargement comme

telecharger.com, clubic.com ... et vous trouvez votre bonheur. Vous récupérez un programme d'installation, vous faites « Suivant », « Suivant », « Suivant », « Terminer » et c'est installé. Parfois, il faut répondre à des questions un peu techniques comme « Dans quel répertoire voulez-vous installer ce programme ? ».

Sous Linux (et notamment sous Ubuntu), ça ne fonctionne pas du tout comme ça : c'est encore plus simple. Mieux : vous allez vite vous rendre compte que c'est un vrai plaisir d'installer de nouveaux programmes et que c'est même un des points forts d'un système comme Ubuntu par rapport à Windows.

Vous allez voir.

Les paquets et leurs dépendances



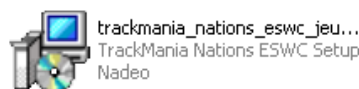
Tout d'abord, il faut savoir que ce dont je vais vous parler ici concerne uniquement les distributions Linux basées sur Debian (je rappelle qu'Ubuntu en fait partie).

En effet, l'installation de programmes fonctionne différemment d'une distribution à une autre. C'est justement une des différences majeures entre les distributions.

Des programmes livrés sous forme de paquets

Sous Windows, vous connaissez ce que l'on appelle des « Programmes d'installation ». En général, il s'agit de fichiers `.exe` à lancer qui s'exécutent et extraient les fichiers du programme dans un dossier `Program Files`.

Exemple : le programme d'installation du jeu Trackmania Nations sous Windows est présenté par la figure suivante.



Sous Ubuntu, on n'a pas de programmes d'installation ; on a ce qu'on appelle des **paquets**.

Un paquet est une sorte de dossier zippé qui contient tous les fichiers du programme. Il se présente sous la forme d'un fichier `.deb`, en référence à **DEB**ian. Il contient toutes les instructions nécessaires pour installer le programme.



Mais alors... un paquet `.deb`, c'est un peu comme un programme d'installation `.exe` sous Windows, non ?

Ça y ressemble, mais en fait ça fonctionne très différemment. Je citerai deux différences notables :

- il y a une gestion des **dépendances** du programme ;
- on n'a pas besoin de faire une recherche sur un moteur de recherche pour trouver un `.deb`. Tous les `.deb` sont rassemblés au même endroit sur un même serveur appelé **dépôt** (*repository*).

Ces deux points méritent plus d'explications.

Dans un premier temps nous allons voir ce que sont les dépendances ; dans un second temps, nous traiterons les dépôts.

Les dépendances, un cauchemar ?

Il est très rare qu'un programme puisse fonctionner seul sous Linux. Très souvent, il utilise d'autres programmes ou d'autres « bouts de programmes » appelés **bibliothèques**. On dit que les programmes dépendent d'autres programmes pour fonctionner : ils ont des **dépendances**.

Par exemple, le programme de dessin The GIMP (équivalent de Photoshop) ne peut pas fonctionner seul. Il dépend de bibliothèques de lecture des images (qui lui disent comment lire une image JPEG) par exemple. Parfois, ces dépendances ont elles-mêmes des dépendances !



Vous avez certainement déjà rencontré un peu ce problème sous Windows. Par exemple, quand vous installez un jeu, on vous dit « Il faut installer Direct X avant de pouvoir jouer à ce jeu ».

Heureusement, le système de paquets Debian est intelligent. Chaque paquet indique de quels autres paquets il dépend. Cela permet au système d'aller récupérer les dépendances manquantes automatiquement si besoin est. Du coup, vous n'avez plus qu'à dire « Je veux installer Nano » et le système ira chercher toutes les dépendances manquantes tout seul !

Les dépôts



Comme je vous l'ai dit un peu plus haut, tous les paquets sont regroupés au sein d'un même endroit appelé dépôt. Il s'agit d'un serveur qui propose **tous** les paquets qui existent (ou presque), ce qui simplifie grandement vos recherches.

Sous Windows, les programmes sont éparpillés aux quatre coins du Net.

Sous Linux, on a décidé de ne pas refaire la même erreur. On a choisi de mettre tout le monde d'accord et de placer tous les programmes (paquets) au même endroit.



Ceci est possible en partie grâce au fait que la plupart des programmes sous Linux sont libres. En effet, les programmes étant « libres », tout le monde est autorisé à les diffuser ; ça ne pose donc aucun problème de les voir tous rassemblés au même endroit.

Sous Windows, la majorité des programmes étant propriétaire, leurs auteurs n'auraient jamais donné leur accord pour que ceux-ci soient diffusés au même endroit (gratuitement, qui plus est).

La notion de dépôt

L'endroit où tous les paquets se trouvent est appelé **dépôt** (*repository* en anglais).



Si tout le monde va chercher ses paquets sur un même dépôt, ça ne risque pas d'engorger le pauvre serveur qui les distribue ?

Bonne remarque, vous avez tout à fait raison. On peut certes mettre un gros serveur avec une grosse bande passante (qui permet à plusieurs centaines de personnes de télécharger en même temps), mais on peut difficilement imaginer que tous les linuxiens de la planète aillent se servir au même endroit au même moment !

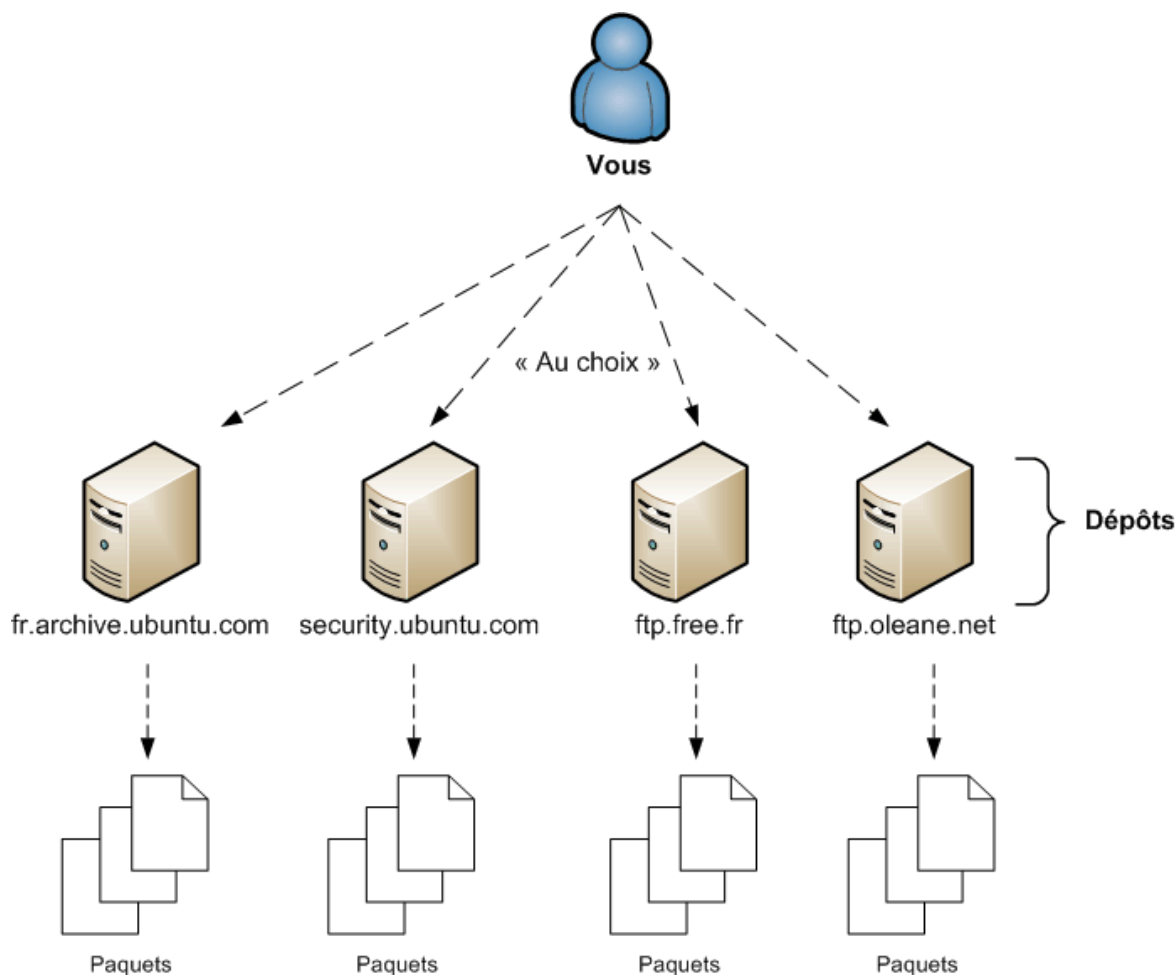
C'est pour cela qu'il existe en fait un grand nombre de dépôts. La plupart proposent exactement les mêmes paquets (les dépôts sont donc des copies les uns des autres).

Certains dépôts spéciaux proposent toutefois des programmes que l'on ne trouve nulle part ailleurs, mais il est rare que l'on ait besoin de se servir dans ces dépôts.



En règle générale, avoir un seul « bon » dépôt suffit à installer la plupart des programmes qui existent sous Linux !

Schématiquement, je représenterais les dépôts comme sur la figure suivante.



C'est donc à vous de choisir le dépôt que vous voulez utiliser. Chacun de ces dépôts est identique : peu importe celui que vous choisissiez, vous devriez retrouver les mêmes paquets.

Comme vous allez probablement beaucoup télécharger depuis votre dépôt, il est conseillé de choisir un serveur qui soit proche de chez vous (sur lequel vous téléchargez suffisamment vite).

En France, par défaut, Ubuntu utilise le dépôt `fr.archive.ubuntu.com`. Ce n'est pas toujours une bonne idée de garder le dépôt par défaut car en cas de nouvelle version d'Ubuntu et de ses logiciels, celui-ci est surchargé et devient alors très lent.

Si vous êtes chez Free, je vous recommande d'utiliser le dépôt de Free.

Si vous êtes chez Wanadoo / Orange, je vous recommande d'utiliser le dépôt Oleanne (appartenant à Orange).

Nous allons voir comment changer de dépôt.

Gérer ses dépôts

Par défaut, quand vous installez Ubuntu, celui-ci utilise les dépôts officiels de la distribution. Seulement, comme je vous l'ai dit plus tôt, ces serveurs risquent d'être souvent encombrés.

Il n'y a aucun mal à utiliser les dépôts officiels, mais il peut être bien de savoir en changer. D'ailleurs, la liste des dépôts que votre ordinateur utilise est stockée dans un fichier. Pour éditer ce fichier, il faut utiliser un éditeur de texte comme... Nano, que l'on a vu au chapitre précédent justement (j'avais tout calculé, qu'est-ce que vous croyez ?).

C'est le moment de mettre en pratique ce que vous avez appris au chapitre précédent. Le fichier à ouvrir qui contient la

liste des dépôts que vous utilisez est :

```
/etc/apt/sources.list
```

Ce fichier ne peut être modifié que par root, l'administrateur de la machine.

Pour que vous puissiez modifier ce fichier, il faut donc passer root.

Pour cela, vous avez deux possibilités. Soit vous faites un `sudo` juste avant :

```
sudo nano /etc/apt/sources.list
```

... et vous modifierez le fichier en tant que root.

Soit vous passez root « définitivement » en faisant `sudo su` d'abord.

Le fichier que vous avez devrait ressembler à ceci :

```
# See http://help.ubuntu.com/community/UpgradeNotes for how to upgrade to
# newer versions of the distribution.

deb http://fr.archive.ubuntu.com/ubuntu/ hardy main restricted
deb-src http://fr.archive.ubuntu.com/ubuntu/ hardy main restricted

## Major bug fix updates produced after the final release of the
## distribution.
deb http://fr.archive.ubuntu.com/ubuntu/ hardy-updates main restricted
deb-src http://fr.archive.ubuntu.com/ubuntu/ hardy-updates main restricted

## N.B. software from this repository is ENTIRELY UNSUPPORTED by the Ubuntu
## team, and may not be under a free licence. Please satisfy yourself as to
## your rights to use the software. Also, please note that software in
## universe WILL NOT receive any review or updates from the Ubuntu security
## team.
deb http://fr.archive.ubuntu.com/ubuntu/ hardy universe
deb-src http://fr.archive.ubuntu.com/ubuntu/ hardy universe

## N.B. software from this repository is ENTIRELY UNSUPPORTED by the Ubuntu
## team, and may not be under a free licence. Please satisfy yourself as to
## your rights to use the software. Also, please note that software in
## multiverse WILL NOT receive any review or updates from the Ubuntu
## security team.
deb http://fr.archive.ubuntu.com/ubuntu/ hardy multiverse
deb-src http://fr.archive.ubuntu.com/ubuntu/ hardy multiverse

## Uncomment the following two lines to add software from the 'backports'
## repository.
## N.B. software from this repository may not have been tested as
## extensively as that contained in the main release, although it includes
## newer versions of some applications which may provide useful features.
## Also, please note that software in backports WILL NOT receive any review
## or updates from the Ubuntu security team.
# deb http://fr.archive.ubuntu.com/ubuntu/ hardy-backports main restricted universe multiverse
# deb-src http://fr.archive.ubuntu.com/ubuntu/ hardy-backports main restricted universe multiverse
```

```
deb http://security.ubuntu.com/ubuntu hardy-security main restricted
deb-src http://security.ubuntu.com/ubuntu hardy-security main restricted
deb http://security.ubuntu.com/ubuntu hardy-security universe
deb-src http://security.ubuntu.com/ubuntu hardy-security universe
deb http://security.ubuntu.com/ubuntu hardy-security multiverse
deb-src http://security.ubuntu.com/ubuntu hardy-security multiverse
```

Les lignes commençant par un `#` sont des lignes de commentaires. Elles seront ignorées.

Normalement, chaque ligne du fichier commence par une de ces deux directives :

- **deb** : pour télécharger la version compilée (binaire) des programmes. C'est ce que vous voudrez faire dans la plupart des cas car c'est la version « prête à l'emploi » ;
- **deb-src** : permet de récupérer le code source du programme. Généralement, vous n'en avez pas besoin, sauf si vous êtes curieux et que vous voulez voir la source d'un programme. C'est l'avantage des logiciels libres de pouvoir consulter la source des programmes !

A priori seules les lignes `deb` nous intéressent. On pourrait même supprimer (ou commenter) les lignes `deb-src`, récupérer les sources n'ayant aucun intérêt pour nous.

Voici une ligne « type » :

```
deb http://fr.archive.ubuntu.com/ubuntu/ hardy universe
```

En premier paramètre, on a l'adresse du dépôt. Ici, le dépôt français par défaut est

```
http://fr.archive.ubuntu.com/ubuntu/ .
```

Ensuite, on a le nom de la version de la distribution qu'on utilise, « hardy » dans ce cas.

Enfin, le dernier paramètre (et tous les paramètres suivants s'il y en a) correspond à la « section » du dépôt dans laquelle vous voulez regarder.

Ouf... c'est un peu compliqué, tout ça.

En fait, la seule chose que vous devriez avoir à faire, c'est remplacer toutes les adresses (http...) par celle du nouveau dépôt que vous voulez utiliser.



Mais comment je connais l'adresse des autres dépôts qui existent ?

Bonne question... à laquelle je ne peux pas répondre.

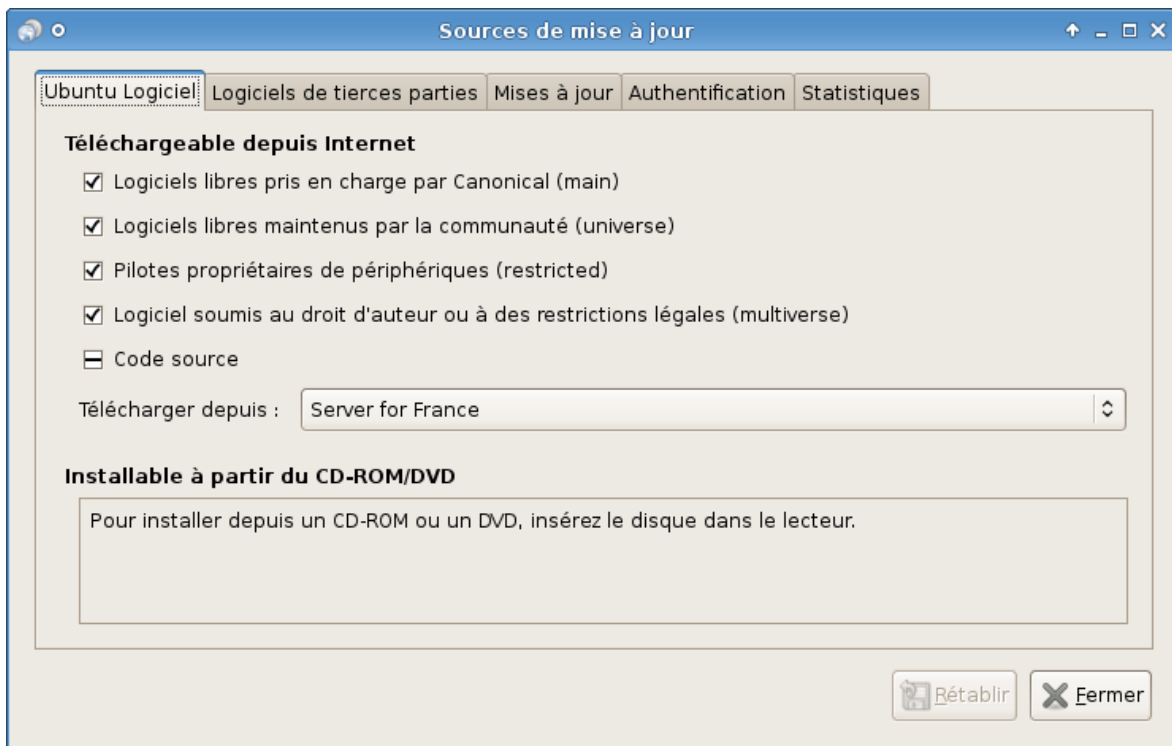
A priori tout le monde peut créer un dépôt, il peut donc très bien exister des dizaines, voire des centaines de dépôts différents que vous pourriez utiliser. Je ne les connais pas tous et je ne vais pas me risquer à dresser une liste ici, mais si vous recherchez « dépôts ubuntu » avec votre moteur de recherche favori, vous devriez trouver des réponses.

Utiliser l'outil graphique

Le plus simple, je pense, est de passer par l'outil graphique fourni par Ubuntu. L'outil en question dépend de votre Ubuntu. Si vous avez :

- **Ubuntu** : allez dans `Système` → `Administration` → `Sources de logiciels` ;
- **KUbuntu** : allez dans `Menu K` → `Système` → `Gestionnaire Adept` → `Adept` → `Gérer les dépôts` ;
- **XUbuntu** : allez dans `Applications` → `Système` → `Sources de mises à jour` .

Par exemple, la fenêtre sous Xubuntu est présentée sur la figure suivante.

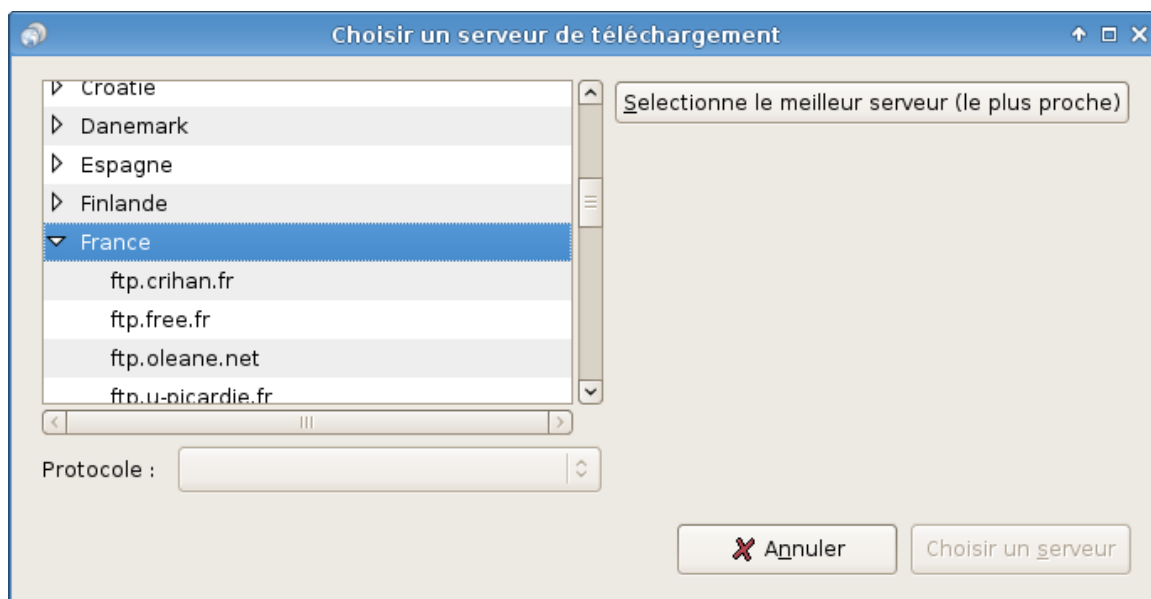


Notez la liste déroulante « Télécharger depuis : Server for France », qui signifie que vous utilisez les dépôts français officiels d'Ubuntu.

Cette liste vous offre le choix pour le moment entre « Main server » (le serveur américain officiel) et le serveur français. Cliquez sur « Autre... » (figure suivante).

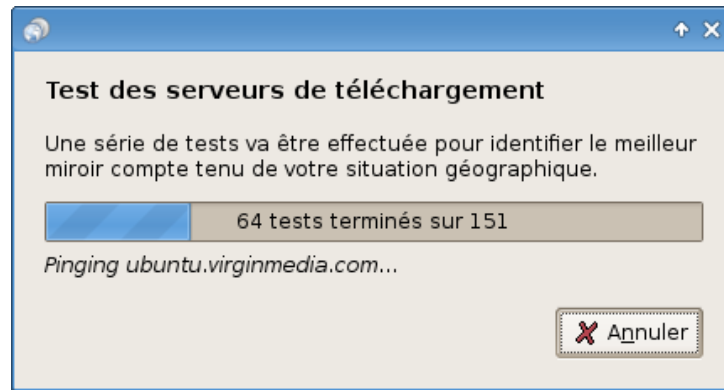


Cette fenêtre recense les dépôts les plus connus regroupés par pays (figure suivante). Certains sont fournis par des universités (u-picardie.fr), d'autres par des FAI (Free, Olean pour Orange...).



Pour savoir lequel est le plus rapide, cliquez sur le bouton « Sélectionner le meilleur serveur (le plus proche) ». Une

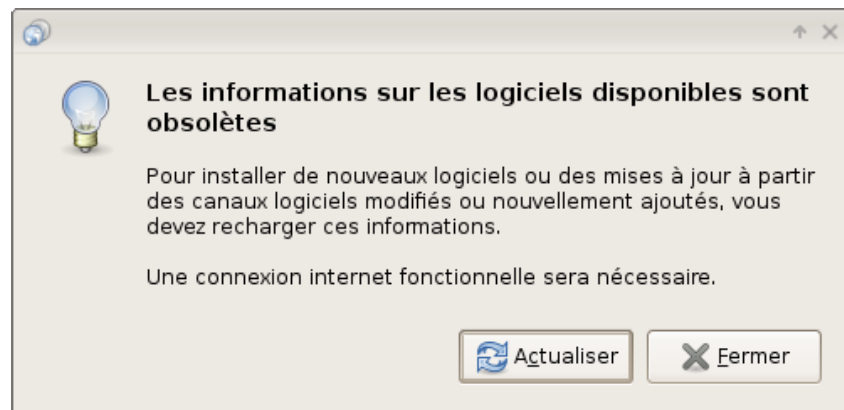
fenêtre va s'ouvrir et tester la vitesse de tous les serveurs (figure suivante).



Le serveur qui sera sélectionné à la fin sera celui que le logiciel aura détecté comme étant le plus rapide pour vous. Cliquez sur « Choisir un serveur » pour valider.

Ensuite, le logiciel vous fera remarquer que votre cache n'est pas à jour. En effet, pour des raisons de performances, Ubuntu avait téléchargé la liste des paquets proposés par l'ancien serveur. Comme vous venez d'en changer, il se peut que la liste des paquets ait changé elle aussi. Il faut récupérer la liste des paquets proposés par le nouveau serveur et la mettre en cache.

Cliquez sur « Actualiser » pour mettre à jour la liste des logiciels disponibles (figure suivante).



Et voilà !

L'outil graphique est plus pratique pour mettre à jour votre fichier `sources.list`, il faut bien le reconnaître.

Maintenant que nous avons choisi notre dépôt, nous sommes prêts à télécharger à gogo !

Les outils de gestion des paquets



Résumons.

Jusqu'ici, nous avons découvert trois nouveaux termes :

- **paquet** : c'est un programme « prêt à l'emploi », l'équivalent des programmes d'installation sous Windows en quelque sorte ;
- **dépendance** : un paquet peut avoir besoin de plusieurs autres paquets pour fonctionner, on dit qu'il a des dépendances ;
- **dépôt** : c'est le serveur sur lequel on va télécharger nos paquets.

Je vous ai proposé de changer de dépôt mais sachez que ce n'est pas obligatoire, vous pouvez vous contenter de celui par défaut.

Sous Ubuntu, on peut utiliser un programme graphique qui gère les paquets pour nous : nous avons justement découvert la logithèque Ubuntu au début de ce cours.

Ici, nous nous intéressons aux manipulations en console. Les deux programmes console de gestion des paquets les plus connus sont :

- `apt-get` ;
- `aptitude` .

Lequel des deux utiliser ?

Le premier est sûrement le plus célèbre ; le second est généralement reconnu comme étant plus efficace lors de la désinstallation de paquets (il supprime aussi les dépendances inutilisées). Cependant `apt-get` sous Ubuntu a évolué aussi et peut supprimer les dépendances inutilisées.

Après, c'est un peu une question d'habitude. Pour ma part, j'ai toujours été habitué à `apt-get` , c'est donc celui que j'utilise et que je vais vous montrer. Que vous utilisiez l'un ou l'autre ne fera pas beaucoup de différence.

Nous devons généralement suivre trois étapes pour télécharger un paquet :

- `apt-get update` (optionnel) : pour mettre notre cache à jour si ce n'est pas déjà fait ;
- `apt-cache search monpaquet` (optionnel) : pour rechercher le paquet que nous voulons télécharger si nous ne connaissons pas son nom exact ;
- `apt-get install monpaquet` : pour télécharger et installer notre paquet.

C'est très simple, vous allez voir.

Nous verrons aussi comment supprimer un paquet et comment mettre tous nos paquets à jour en une seule commande !

apt-get update : mettre à jour le cache des paquets



Commençons par la mise à jour du cache des paquets (`apt-get update`).

Cela correspond à télécharger la nouvelle liste des paquets proposés par le dépôt.

Toutefois, il n'est pas nécessaire de mettre à jour son cache à chaque fois que l'on veut télécharger un paquet.



Ah bon ? Comment je sais si je dois mettre à jour mon cache, alors ?

Il y a deux cas où vous avez besoin de le mettre à jour :

- quand vous changez ou ajoutez un dépôt à votre liste de dépôts ;
- quand vous n'avez pas mis à jour votre cache depuis un moment (quelques semaines).

Pour mettre à jour votre cache, tapez ceci dans la console **en tant que root** :

```
apt-get update
```



Pensez à rajouter un `sudo` si vous n'êtes pas déjà root.

Après avoir tapé cette commande, vous allez automatiquement télécharger la dernière liste des paquets proposés par vos dépôts :

```
root@mateo21-desktop:~# apt-get update
Réception de : 1 http://wine.budgetdedicated.com hardy Release.gpg [191B]
Ign http://wine.budgetdedicated.com hardy/main Translation-fr
Atteint http://wine.budgetdedicated.com hardy Release
Atteint ftp://ftp.free.fr hardy Release.gpg
Ign http://wine.budgetdedicated.com hardy/main Packages
Atteint ftp://ftp.free.fr hardy/restricted Translation-fr
Atteint http://wine.budgetdedicated.com hardy/main Sources
Atteint ftp://ftp.free.fr hardy/main Translation-fr
Atteint http://wine.budgetdedicated.com hardy/main Packages
Atteint ftp://ftp.free.fr hardy/universe Translation-fr
Atteint ftp://ftp.free.fr hardy/multiverse Translation-fr
Atteint ftp://ftp.free.fr hardy-updates Release.gpg
Réception de : 2 ftp://ftp.free.fr hardy-updates/restricted Translation-fr
Ign ftp://ftp.free.fr hardy-updates/restricted Translation-fr
Réception de : 3 ftp://ftp.free.fr hardy-updates/main Translation-fr
Ign ftp://ftp.free.fr hardy-updates/main Translation-fr
Réception de : 4 ftp://ftp.free.fr hardy-security Release.gpg [191B]
Réception de : 5 ftp://ftp.free.fr hardy-security/restricted Translation-fr
Ign ftp://ftp.free.fr hardy-security/restricted Translation-fr
Réception de : 6 ftp://ftp.free.fr hardy-security/main Translation-fr
Ign ftp://ftp.free.fr hardy-security/main Translation-fr
Réception de : 7 ftp://ftp.free.fr hardy-security/universe Translation-fr
Ign ftp://ftp.free.fr hardy-security/universe Translation-fr
Réception de : 8 ftp://ftp.free.fr hardy-security/multiverse Translation-fr
Ign ftp://ftp.free.fr hardy-security/multiverse Translation-fr
Réception de : 9 ftp://ftp.free.fr hardy Release [57,2kB]
Réception de : 10 ftp://ftp.free.fr hardy-updates Release [50,9kB]
Réception de : 11 ftp://ftp.free.fr hardy-security Release [50,9kB]
Atteint ftp://ftp.free.fr hardy/restricted Packages
Atteint ftp://ftp.free.fr hardy/main Packages
Atteint ftp://ftp.free.fr hardy/restricted Sources
Atteint ftp://ftp.free.fr hardy/universe Packages
Atteint ftp://ftp.free.fr hardy/universe Sources
Atteint ftp://ftp.free.fr hardy/multiverse Packages
Atteint ftp://ftp.free.fr hardy/multiverse Sources
Atteint ftp://ftp.free.fr hardy-updates/restricted Packages
Atteint ftp://ftp.free.fr hardy-updates/main Packages
Atteint ftp://ftp.free.fr hardy-updates/restricted Sources
Réception de : 12 ftp://ftp.free.fr hardy-security/restricted Packages [5990B]
Réception de : 13 ftp://ftp.free.fr hardy-security/main Packages [120kB]
Réception de : 14 ftp://ftp.free.fr hardy-security/restricted Sources [956B]
Réception de : 15 ftp://ftp.free.fr hardy-security/universe Packages [78,6kB]
Réception de : 16 ftp://ftp.free.fr hardy-security/universe Sources [11,8kB]
Réception de : 17 ftp://ftp.free.fr hardy-security/multiverse Packages [5395B]
Réception de : 18 ftp://ftp.free.fr hardy-security/multiverse Sources [1042B]
382ko réceptionnés en 7s (50,1ko/s)
Lecture des listes de paquets... Fait
```

Voilà : ça fait un peu peur la première fois mais en général, c'est assez rapide.

Je vous rappelle que vous n'avez pas besoin d'exécuter cette commande à chaque fois que vous voulez installer un paquet mais seulement de temps en temps pour être sûrs d'avoir la liste la plus à jour possible.

apt-cache search : rechercher un paquet



À moins que vous ne connaissiez déjà le nom exact du paquet que vous voulez, il va falloir effectuer une petite recherche.

On utilise pour cela la commande suivante :

```
apt-cache search votrerecherche
```

Cette commande effectue une recherche de paquet dans votre cache. Cela évite d'avoir à aller sur Internet pour faire la recherche, ce qui aurait été lent.

Bon ! Que veut-on télécharger ? Un éditeur de texte ? Un navigateur ?

Allez : pour s'amuser, je vous propose de rechercher un jeu, par exemple un jeu de casse-briques (*breakout* en anglais).

Faites la recherche suivante :

```
root@mateo21-desktop:~# apt-cache search breakout
briquolo - Fast paced 3d Breakout
briquolo-data - Fast paced 3d Breakout data files
circuslinux - The clowns are trying to pop balloons to score points!
circuslinux-data - data files for circuslinux
gnome-breakout - Clone of the classic game Breakout, written for GNOME
lbreakout2 - A ball-and-paddle game with nice graphics
lbreakout2-data - A ball-and-paddle game with nice graphics (DATA FILES)
libfreebob0 - FreeBoB API
libfreebob0-dev - FreeBoB API - development files
tecnoballz - breaking block game ported from the Amiga platform
```

La commande `apt-cache search breakout` a listé tous les paquets qui avaient un rapport avec les casse-briques. À gauche vous avez le nom du paquet, à droite une courte description.



Si vous voulez une plus ample description d'un paquet, utilisez `apt-cache show nomdupaquet`. Exemple :
`apt-cache show lbreakout2` .

apt-get install : installer un paquet



Pour ma part, j'aime beaucoup le jeu `lbreakout2` (figure suivante).

Je vous propose donc de le télécharger, ce qui se fait très simplement (**toujours en tant que root**, rajoutez un `sudo` devant la commande si vous n'êtes pas déjà root) :

```
apt-get install lbreakout2
```



La commande `apt-get install` attend que vous lui donniez le nom du paquet à installer.



Astuce : vous pouvez installer plusieurs paquets d'un coup en les listant un à un :

```
apt-get install paquet1 paquet2 paquet3
```

Essayons donc d'installer `lb breakout2` :

```
root@mateo21-desktop:~# apt-get install lb breakout2
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances
Lecture de l'information d'état... Fait
Les paquets supplémentaires suivants seront installés :
  lb breakout2-data libSDL-mixer1.2 libsmpeg0
Les NOUVEAUX paquets suivants seront installés :
  lb breakout2 lb breakout2-data libSDL-mixer1.2 libsmpeg0
0 mis à jour, 4 nouvellement installés, 0 à enlever et 153 non mis à jour.
Il est nécessaire de prendre 2943ko dans les archives.
Après dépaquetage, 5358ko d'espace disque supplémentaires seront utilisés.
Souhaitez-vous continuer [O/n] ? O
Réception de : 1 ftp://ftp.free.fr/feisty/main/libsmpeg0 0.4.5+cvcs20030824-1.9build1 [105kB]
Réception de : 2 ftp://ftp.free.fr/feisty/main/libSDL-mixer1.2 1.2.6-1.1build1 [145kB]
Réception de : 3 ftp://ftp.free.fr/feisty/universe/lb breakout2-data 2.5.2-2.1ubuntu1 [2444kB]
Réception de : 4 ftp://ftp.free.fr/feisty/universe/lb breakout2 2.5.2-2.1ubuntu1 [249kB]
2943ko réceptionnés en 6s (484ko/s)
Sélection du paquet libsmpeg0 précédemment désélectionné.
(Lecture de la base de données... 123350 fichiers et répertoires déjà installés.)
Dépaquetage de libsmpeg0 (à partir de .../libsmpeg0_0.4.5+cvcs20030824-1.9build1_amd64.deb) ...
```

```

Sélection du paquet libSDL-mixer1.2 précédemment désélectionné.
Dépaquetage de libSDL-mixer1.2 (à partir de ../libSDL-mixer1.2_1.2.6-1.1build1_amd64.deb) ...
Sélection du paquet lbreakout2-data précédemment désélectionné.
Dépaquetage de lbreakout2-data (à partir de ../lbreakout2-data_2.5.2-2.1ubuntu1_all.deb) ...
Sélection du paquet lbreakout2 précédemment désélectionné.
Dépaquetage de lbreakout2 (à partir de ../lbreakout2_2.5.2-2.1ubuntu1_amd64.deb) ...
Paramétrage de libsmpeg0 (0.4.5+cvs20030824-1.9build1) ...

Paramétrage de libSDL-mixer1.2 (1.2.6-1.1build1) ...

Paramétrage de lbreakout2-data (2.5.2-2.1ubuntu1) ...
Paramétrage de lbreakout2 (2.5.2-2.1ubuntu1) ...

```



Si vous obtenez le message d'erreur suivant :

```

E: Impossible d'ouvrir le fichier verrou /var/lib
/dpkg/lock - open (13 Permission non accordée)
E: Unable to lock the administration directory (/var/lib/dpkg/), are you root?

```

... c'est que vous n'avez pas exécuté la commande en tant que root. Pensez à passer root (en utilisant `sudo`) avant de l'exécuter.

Il s'en est passé des choses, dites donc !

Comme vous pouvez le voir, au début `apt-get` a vérifié si le paquet existait et les dépendances dont il avait besoin. Il se trouve que `lbreakout2` avait besoin de dépendances qu'on n'a pas, comme `libSDLmixer-1.2` par exemple. C'est pour cela qu'on nous a demandé confirmation au milieu :

```
Souhaitez-vous continuer [O/n] ?
```

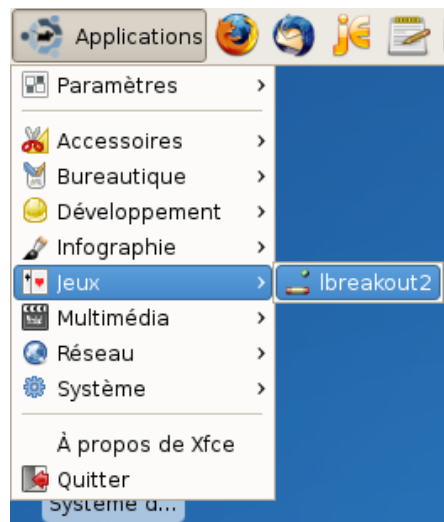
Répondez par un « O » majuscule (comme « Oui ») et tapez `Entrée` pour que l'installation se poursuive.

C'est alors que la magie d' `apt-get` opère : le programme va aller télécharger tout seul le paquet sur le dépôt ainsi que toutes les dépendances dont il a besoin et que nous n'avons pas.

Puis il « dépaquète » les fichiers qui étaient contenus à l'intérieur du paquet, les installe et effectue les paramétrages tout seul.

Ce qui est fort là-dedans, c'est qu' `apt-get` ne vous demande rien ! Il installe tout ce qu'il faut tout seul, dans les bons répertoires, et crée même le raccourci pour lancer le jeu dans la section « Jeux » !

Vérifiez par vous-mêmes. Le menu des applications d'Ubuntu comporte désormais une section `Jeux` (qu'il a créée si elle n'existait pas) et possède un raccourci vers le jeu que nous venons d'installer (figure suivante).



C'est bon, vous pouvez jouer !

Recommencez l'opération autant de fois que vous voulez, tous les paquets que vous devriez voir sont des logiciels libres que vous pouvez télécharger à volonté.

Sur la figure suivante, vous pouvez voir OpenArena, basé sur le célèbre jeu Quake III Arena dont le code source est devenu libre.



Pour l'obtenir, c'est très simple :

```
apt-get install openarena
```

C'est aussi simple que ça. Toutefois il est fortement conseillé d'avoir installé les pilotes de votre carte graphique avant d'y jouer.

apt-get autoremove : supprimer un paquet



Si vous voulez désinstaller un paquet, vous pouvez utiliser la commande `apt-get remove` :

```
apt-get remove lbreakout2
```

Le paquet sera alors désinstallé de votre ordinateur.

Toutefois, cela ne supprime pas les dépendances du paquet devenues inutiles. Pour demander à `apt-get` de supprimer **aussi** les dépendances inutiles, on utilise `autoremove` :

```
apt-get autoremove lbreakout2
```

Faisons un `autoremove`, comme ça nous serons sûrs de libérer un maximum d'espace disque.

```
root@mateo21-desktop:~# apt-get autoremove lbreakout2
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances
Lecture de l'information d'état... Fait
Les paquets suivants ont été automatiquement installés mais ne sont plus nécessaires :
  libSDL-mixer1.2 libsmpeg0
Les paquets suivants seront ENLEVÉS :
  lbreakout2 lbreakout2-data libSDL-mixer1.2 libsmpeg0
0 mis à jour, 0 nouvellement installés, 4 à enlever et 0 non mis à jour.
Il est nécessaire de prendre 0o dans les archives.
Après dépaquetage, 5358ko d'espace disque seront libérés.
Souhaitez-vous continuer [O/n] ?
```

`apt-get` analyse le paquet, vérifie les dépendances qu'il peut supprimer sans risque (c'est-à-dire les dépendances qui ne sont plus utilisées par aucun autre paquet) et calcule la taille d'espace disque libéré (un peu plus de 5 Mo dans mon cas).

Faites « O » pour confirmer que vous voulez bien désinstaller le paquet et ses dépendances.

Vous voyez alors les paquets se faire désinstaller par `apt-get` :

```
Suppression de lbreakout2 ...
Suppression de lbreakout2-data ...
Suppression de libSDL-mixer1.2 ...
Suppression de libsmpeg0 ...
```

C'est fini, le paquet et ses dépendances sont proprement désinstallés. :-)

apt-get upgrade : mettre à jour tous les paquets



Une autre fonctionnalité particulièrement géniale d' `apt-get` est sa capacité à mettre à jour tous les paquets installés sur votre système d'un seul coup. Le programme ira chercher les nouvelles versions de tous vos programmes et les mettra à jour si une nouvelle version est disponible :

```
apt-get upgrade
```



Pensez à faire un `apt-get update` pour mettre à jour le cache des paquets sur votre machine avant de lancer un `upgrade` .

En effet, `apt-get` compare la version de vos paquets installés avec ceux présents dans le cache. Si votre cache est « ancien », `apt-get` se dira « Oh bah, il n'y a rien de nouveau, pas besoin d'une mise à jour ». Veillez donc à

faire régulièrement un `apt-get update` pour être 100 % sûrs que votre cache est à jour.

On vous demandera une confirmation après avoir listé tous les paquets qui ont besoin d'une mise à jour.

Vous n'avez pas besoin de faire autre chose. Tous vos paquets installés seront mis à jour (ça peut être un peu long, par contre).

Si tous vos paquets sont déjà dans leur version la plus récente, vous verrez le message suivant :

```
root@mateo21-desktop:~# apt-get upgrade
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances
Lecture de l'information d'état... Fait
0 mis à jour, 0 nouvellement installés, 0 à enlever et 0 non mis à jour.
```


Tous ces « 0 » de la dernière ligne nous indiquent que rien ne s'est passé car il n'y avait rien à faire.

Cela signifie que votre système est à jour.



Il est conseillé de faire un `apt-get upgrade` régulièrement pour avoir le système le plus à jour possible. Cela vous permet de bénéficier des dernières fonctionnalités des logiciels, mais cela corrige aussi les failles de sécurité qui auraient pu être découvertes dans les programmes (et on en trouve tous les jours, même dans les logiciels libres !).

En résumé

- La plupart des distributions Linux proposent un moyen centralisé de télécharger et d'installer des logiciels facilement. L'installation de programmes y est généralement beaucoup plus simple que sous Windows !
 - Sous Ubuntu, on peut utiliser des interfaces graphiques pour télécharger et installer des logiciels. En console, on appelle au programme `apt-get`.
- Le professeur**
-  **Mathieu Nebra**
Entrepreneur à plein temps, auteur à plein temps et co-fondateur d'OpenClassrooms :o)
- Il télécharge les programmes depuis des serveurs (fournis par Ubuntu, votre fournisseur d'accès ou une université) qui font office de dépôts.

- `apt-get update` met à jour la liste des programmes (appelés *paquets*) qui existent.
 - `apt-cache search` permet de rechercher dans la liste des paquets.
- Découvrez aussi ce cours en...**



`apt-get install` télécharge et installe un paquet.

`apt-get upgrade` met à jour tous les paquets installés.

Livre

PDF

`apt-get autoremove` permet de supprimer un paquet.



Que pensez-vous de ce cours ?

OPENCLASSROOMS

Qui sommes-nous ?

J'AI TERMINÉ CE CHAPITRE ET JE PASSE AU SUIVANT

Financements

[← NANO, L'ÉDITEUR DE TEXTE DU DÉBUTANT](#)

Expérience de formation

[RTFM : LISEZ LE MANUEL !](#)



Forum

Blog [↗](#)

Presse [↗](#)

OPPORTUNITÉS

Nous rejoindre [↗](#)

Devenir mentor [↗](#)

Devenir coach carrière [↗](#)

AIDE



FAQ

POUR LES ENTREPRISES

Former et recruter

EN PLUS

Boutique [↗](#)

Mentions légales

Conditions générales d'utilisation

Politique de protection des données personnelles

Cookies

Accessibilité



Français



Télécharger dans
l'App Store