

[Accueil](#) > [Cours](#) > [Reprenez le contrôle à l'aide de Linux !](#) > Compiler un programme depuis les sources

Reprenez le contrôle à l'aide de Linux !

 30 heures  Facile

Mis à jour le 29/06/2021



Compiler un programme depuis les sources

Nous avons découvert dans un chapitre précédent combien il était facile d'installer de nouveaux programmes sous Ubuntu à l'aide de la commande `apt-get`. Cette technique permet de télécharger et installer la grande majorité des programmes.

Cependant, il arrive parfois qu'il soit nécessaire d'installer un programme manuellement car il n'apparaît pas dans `apt-get`. Dans ce cas, il faut récupérer les sources du programme et les compiler soi-même pour créer un exécutable !

Cette opération peut se révéler assez complexe dans certains cas. Il nous faudra télécharger les sources, les extraire d'une archive gzippée et les compiler manuellement. Je vous propose dans ce chapitre de mettre en pratique vos connaissances en découvrant pas à pas la compilation d'un programme.

Essayez d'abord de trouver un paquet .deb



La plupart des programmes dont vous aurez besoin sous Ubuntu sont référencés dans des dépôts et accessibles via une simple commande : `apt-get`. Toutefois, certains programmes récents ou encore en développement ne sont pas disponibles via `apt-get` (c'est le cas des programmes en cours de développement ou de ceux qui ne sont pas encore assez connus pour être intégrés aux dépôts officiels d'Ubuntu).

Dans un tel cas, les choses se corsent nettement. Sous Windows, nous avons l'habitude de nous

rendre sur le site web du logiciel et de télécharger le `.exe` d'installation. Or, les développeurs qui écrivent des programmes pour Linux ne créent presque jamais de gestionnaires d'installation comme on en voit sous Windows.



Pourquoi ? Le concept de programme d'installation n'existe-t-il pas sous Linux ?

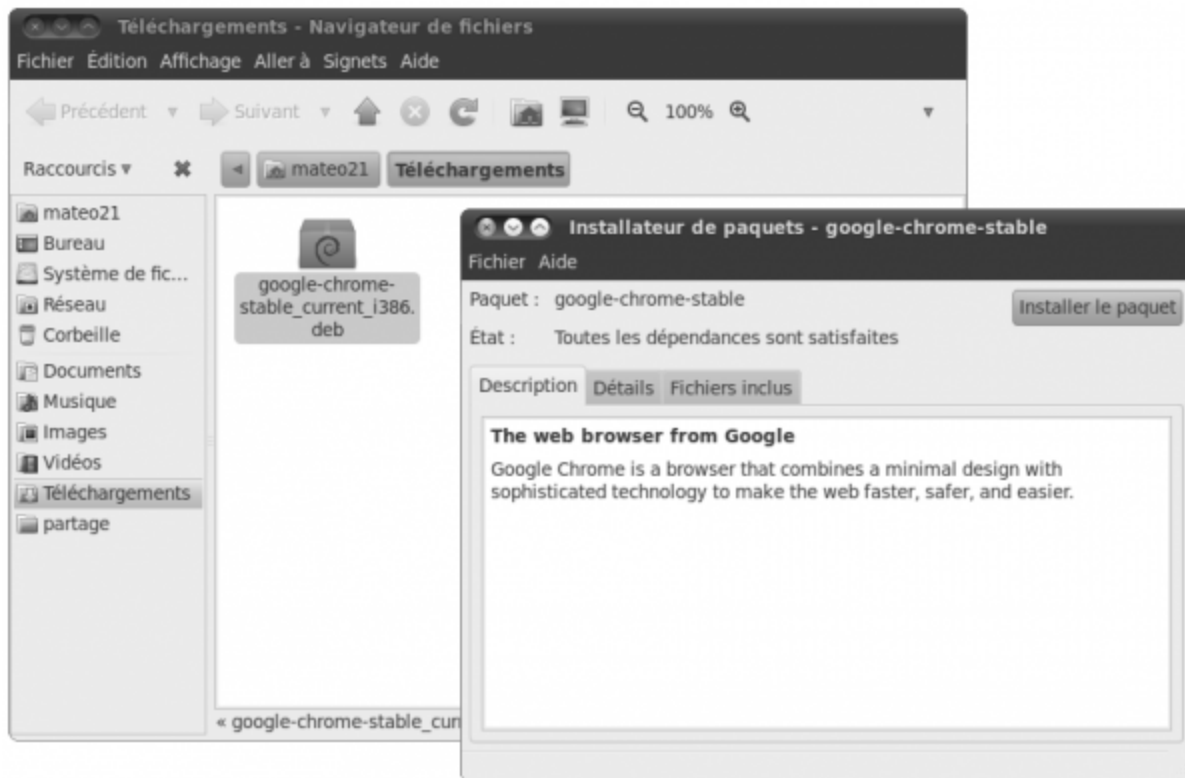
En fait, ce concept existe mais on l'évite. Pourquoi ? La raison est qu'il existe une grande diversité de distributions Linux et d'architectures d'ordinateurs (selon le type de processeur par exemple). Cette diversité est une des grandes forces de Linux, mais il est du coup presque impossible de proposer un programme d'installation qui convienne à tout le monde et qui pourra s'installer sur toutes les machines. Il faudrait créer autant de programmes d'installation qu'il existe de types de machines différents !

Quand `apt-get` ne propose pas le programme que l'on recherche, il est parfois possible de trouver sur le site web du logiciel un paquetage `.deb`. C'est en quelque sorte l'équivalent du programme d'installation, mais celui-ci est spécifique à Debian et à ses distributions dérivées (dont fait partie Ubuntu). Les `.deb` ne fonctionnent pas sur les distributions utilisant d'autres outils ; Red Hat utilise des `.rpm` par exemple.

Notez que le programme `alien` est capable de convertir un `.rpm` en `.deb` au besoin.

Si, par bonheur, vous trouvez le `.deb` du programme que vous souhaitez installer, téléchargez-le et double-cliquez dessus. Essayons par exemple de récupérer de cette façon [Google Chrome sur le site web de Google](#).

Une fois le `.deb` téléchargé, double-cliquez dessus. Une fenêtre apparaît pour vous proposer d'installer le logiciel (figure suivante).



Si aucune erreur n'apparaît, vous avez de la chance, vous pouvez procéder à l'installation. Sinon, cela signifie :

- soit que vous avez téléchargé un `.deb` ne correspondant pas à votre machine. Vérifiez que vous n'avez pas pris une version 32 bits au lieu de 64 bits (ou inversement) ;
- soit qu'il vous manque des dépendances pour pouvoir installer convenablement le programme. Et là, cela peut vite devenir un casse-tête ! Il faut d'abord installer le programme manquant avant d'aller plus loin.



Souvenez-vous : chaque programme dépend de l'installation d'autres programmes, comme nous l'avions découvert précédemment. Un outil comme `apt-get` permet de télécharger automatiquement les dépendances du programme, ce qui simplifie grandement les choses.

Si même le packaging `.deb` n'est pas disponible, il ne reste alors qu'une solution : récupérer le *code source* du programme et le compiler soi-même. On peut ainsi créer un exécutable spécialement optimisé pour sa machine.

L'exécutable est l'équivalent du `.exe` de Windows, même s'il n'a en général pas d'extension sous Linux.

Quand il n'y a pas d'autre solution : la compilation



Si le programme que vous recherchez n'est pas dans les dépôts (`apt-get`) et que vous ne parvenez pas non plus à trouver de `.deb` prêt à l'emploi sur le web, vous allez devoir le compiler depuis ses sources.

Qu'est-ce que la compilation ?

La **compilation** est un procédé qui permet de transformer le code source d'un programme en un exécutable que l'on peut utiliser. Le code source correspond en quelque sorte aux ingrédients d'une recette (les œufs, la farine...) et l'exécutable au gâteau final. Dans cette métaphore, la compilation correspondrait à la cuisson du gâteau. :-)

Étant donné que la plupart des programmes sous Linux sont libres, nous avons la chance de pouvoir récupérer leurs sources et donc de pouvoir en compiler une version propre à notre machine.

Les étapes de la compilation peuvent varier d'un programme à un autre. Certains sont assez complexes et nécessitent plusieurs préparatifs. Dans ce cas, il faut suivre les instructions indiquées sur le site web du logiciel pour savoir comment compiler (instructions qui sont, bien souvent, en anglais).

Compilation d'un programme pas à pas

Pour compiler des programmes, vous aurez besoin avant toute chose d'installer les outils de compilation. Pour cela, rien de plus simple, il suffit d'installer le paquet `build-essential` :

```
sudo apt-get install build-essential
```

Ceci étant fait, nous pouvons à présent nous intéresser à la compilation proprement dite.

Ici, je vous propose d'apprendre à compiler un petit programme assez simple : `htop` . Il s'agit d'un outil alternatif à `top` , qui permet de voir la liste des programmes en cours d'exécution. Cela sera l'occasion de découvrir les principales commandes de compilation qui vous serviront pour installer la plupart des logiciels.

Notez qu'on le retrouve dans les dépôts via `apt-get` , mais nous allons tout de même essayer de le compiler manuellement pour nous entraîner.

La première étape consiste à se rendre sur [le site web du logiciel htop](#). Une recherche sur le web devrait vous y amener rapidement.

À partir de là, il est indispensable de savoir lire l'anglais. Recherchez sur le site la section « Downloads », puis, sur la page des téléchargements, recherchez les sources. Vous devriez finalement arriver sur une page qui vous propose de [télécharger les dernières sources du programme](#).

Vous allez télécharger une archive compressée `.tar.gz`. Vous connaissez la commande pour extraire ce type d'archive, alors allez-y !

```
tar zxvf htop-0.8.3.tar.gz
```

On peut maintenant se rendre dans le dossier où les fichiers sources ont été décompressés :

```
cd htop-0.8.3
```

Si vous listez le contenu de ce répertoire, vous allez être surpris : il y a beaucoup de fichiers ! Heureusement, vous n'avez pas à vous en préoccuper.

Pour le moment, un seul programme nous intéresse : `configure`. Exécutez-le comme suit :

```
./configure
```

`configure` est un programme qui analyse votre ordinateur et qui vérifie si tous les outils nécessaires à la compilation du logiciel que vous souhaitez installer sont bien présents. Son exécution peut prendre du temps car il effectue de nombreux tests :

```
$ ./configure
checking for a BSD-compatible install... /usr/bin/install -c
checking whether build environment is sane... yes
checking for a thread-safe mkdir -p... /bin/mkdir -p
checking for gawk... gawk
checking whether make sets $(MAKE)... yes
checking for gcc... gcc
checking for C compiler default output file name... a.out
checking whether the C compiler works... yes
checking whether we are cross compiling... no
checking for suffix of executables...
checking for suffix of object files... o
checking whether we are using the GNU C compiler... yes
checking whether gcc accepts -g... yes
checking for gcc option to accept ISO C89... none needed
checking for style of include used by make... GNU
checking dependency style of gcc... gcc3

[...]
```

Un des premiers éléments qu'il va vérifier est la présence du compilateur (`checking for gcc...`) que vous avez normalement dû installer un peu plus tôt avec le paquet `build-essential`.

Patientez le temps de l'exécution de `configure`. Celui-ci va vous indiquer si tout est prêt pour une compilation ou non.

Malheureusement, il arrivera fréquemment que `configure` affiche une erreur en raison d'un manque de dépendances. Dans notre cas, il devrait afficher une erreur comme celle-ci :

```
checking for sys/time.h... yes
checking for unistd.h... (cached) yes
checking curses.h usability... no
checking curses.h presence... no
checking for curses.h... no
configure: error: missing headers:  curses.h
```

L'erreur (sur la dernière ligne) indique en anglais « `missing headers: curses.h` ». C'est là que les choses se corsent : il faut installer l'élément manquant, en l'occurrence ces fameux *headers* de `curses.h`. Si vous n'êtes pas programmeurs, vous n'avez probablement aucune idée de ce dont il s'agit.

La technique la plus efficace consiste à effectuer une recherche de la ligne d'erreur sur le web, accompagnée de préférence du mot-clé « ubuntu ». Lancez donc une recherche de « `configure: error: missing headers: curses.h ubuntu` ».

Une [recherche de ces mots clés](#) devrait généralement vous afficher des résultats qui vous dirigeront bien souvent sur des forums anglophones.

Il faut alors faire preuve de patience et ne pas avoir peur de lire un peu d'anglais. 🤔

L'information à chercher est le nom du *paquet* manquant que vous devez installer. En lisant les forums, vous devriez finir par trouver le nom du paquet que vous recherchez : `libncurses5-dev`. En l'occurrence, il suffit d'installer ce paquet via `apt-get` pour ne plus avoir l'erreur indiquée dans `configure`.

```
sudo apt-get install libncurses5-dev
```

Une fois le paquet installé, relancez `configure` et croisez les doigts pour que l'erreur disparaisse.



Cela n'a pas été scientifiquement prouvé, mais croiser les doigts peut augmenter vos chances de réussite. La magie vaudou est aussi un bon moyen de se sortir des situations compliquées, mais elle est à réserver aux utilisateurs expérimentés. 🤞

```
./configure
```

Si `configure` n'affiche plus la même erreur, vous avez gagné (pour le moment). Il reste maintenant deux possibilités :

- soit vous avez une nouvelle erreur et vous devrez la résoudre de la même manière : en effectuant une recherche sur l'internet pour comprendre ce qui ne va pas. Le plus souvent, il suffira d'installer le paquet manquant avec `apt-get` ;
- soit vous n'avez pas d'erreur et `configure` parvient jusqu'à son terme. Victoire !

Si tout va bien, `configure` n'affichera pas d'erreur. Vous devriez voir s'afficher des lignes similaires à celles-ci :

```
configure: creating ./config.status
config.status: creating plpa-1.1/Makefile
config.status: creating plpa-1.1/src/Makefile
config.status: creating Makefile
config.status: creating htop.1
config.status: creating config.h
config.status: config.h is unchanged
config.status: creating plpa-1.1/src/plpa_config.h
config.status: creating plpa-1.1/src/plpa.h
config.status: executing depfiles commands
```

Le programme est prêt à être compilé ! Rassurez-vous, le plus dur est derrière vous. :-)

Il suffit maintenant de lancer la compilation à l'aide d'une commande toute simple :

```
make
```

Durant la compilation, des lignes barbares s'afficheront dans votre console. Vous ne devriez pas avoir à vous en préoccuper, tous les problèmes ayant normalement été détectés auparavant par `configure` .



La compilation d'un programme peut prendre du temps ; tout dépend de la taille de celui-ci. Il est ainsi bien plus rapide et plus simple de compiler htop que Firefox par exemple.

Une fois la compilation terminée, l'exécutable devrait avoir été créé. Il ne reste plus qu'à l'installer, c'est-à-dire à le copier dans le bon répertoire. Là encore, vous n'avez pas à vous poser beaucoup de questions. Exécutez la commande suivante :

```
sudo make install
```

Il faut être « root » pour cette opération (d'où le `sudo`) car le programme va être copié dans des répertoires système.

Une fois que cela est fait, le programme est installé ! Nous pouvons à présent exécuter `htop` en tapant le nom de la commande :

```
htop
```

The screenshot shows the htop interface. At the top, system statistics are displayed: CPU usage at 3.9%, memory usage at 230/1002MB, and swap usage at 0/567MB. System tasks are 226 total with 1 running. Load averages are 0.08, 0.21, and 0.18. Uptime is 08:38:17.

PID	USER	PRI	NI	VIRT	RES	SHR	S	CPU%	MEM%	TIME+	Command
11818	mateo21	20	0	2576	1280	992	R	1.0	0.1	0:07.87	htop
1147	root	20	0	3608	1328	1148	S	0.0	0.1	0:08.59	hald-addon-storage
1013	root	20	0	49880	24488	8988	S	0.0	2.4	0:19.53	/usr/bin/X :0 -nr
6321	mateo21	20	0	248M	40968	26784	S	0.0	4.0	0:01.79	/opt/google/chrome
6359	mateo21	20	0	112M	28512	13368	S	0.0	2.8	0:02.05	/opt/google/chrome
6346	mateo21	20	0	120M	18064	10776	S	0.0	1.8	0:00.42	/opt/google/chrome
1467	mateo21	20	0	50080	13396	10228	S	0.0	1.3	0:11.67	gnome-terminal
6355	mateo21	20	0	248M	40968	26784	S	0.0	4.0	0:00.59	/opt/google/chrome
1290	mateo21	20	0	97472	11808	9636	S	0.0	1.2	0:01.26	metacity --replace
778	nobody	20	0	54116	1104	808	S	0.0	0.1	0:00.97	/usr/bin/memcached
1341	root	20	0	5176	872	604	S	0.0	0.1	0:04.21	udisks-daemon: pol
837	root	20	0	5976	728	520	S	0.0	0.1	0:00.74	/usr/sbin/VBoxServ
539	messageb	20	0	3224	1552	784	S	0.0	0.2	0:00.48	dbus-daemon --syst
1359	mateo21	20	0	16948	2348	1960	S	0.0	0.2	0:01.29	/usr/lib/gvfs/gvfs
1381	mateo21	20	0	32492	13892	11076	S	0.0	1.4	0:00.62	/usr/lib/gnome-pan
1283	mateo21	20	0	20668	8196	6640	S	0.0	0.8	0:00.44	gnome-power-manage
1372	mateo21	20	0	44260	13404	10396	S	0.0	1.3	0:01.27	/usr/lib/gnome-pan

At the bottom, function key shortcuts are listed: F1Help, F2Setup, F3Search, F4Invert, F5Tree, F6SortBy, F7Nice, F8Nice, F9Kill, F10Quit.

Si vous souhaitez désinstaller le programme, il suffit d'exécuter cette commande depuis le répertoire où vous l'avez compilé :

```
sudo make uninstall
```



Vous pouvez sans problème supprimer le répertoire contenant les fichiers sources (celui depuis lequel vous avez compilé). Toutefois, il ne sera alors plus possible de lancer la commande de désinstallation.

En résumé

- La plupart des programmes peuvent être installés facilement avec la commande `apt-get` .
- Certains programmes ne peuvent pas être installés via `apt-get` car ils ne sont pas

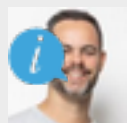
référencés dans les dépôts d'Ubuntu. Dans ce cas, on peut rechercher sur le web un paquet `.deb` du programme, sous réserve qu'il existe.

- Si la solution précédente échoue, on n'a pas d'autre choix que de compiler le programme à partir de ses sources. Cela consiste à effectuer les opérations suivantes dans l'ordre :

1. télécharger les sources du programme sur le web (souvent archivées au format `.tar.gz`) ;
2. décompresser l'archive (`tar zxvf archive.tar.gz`) ;
3. exécuter `./configure` et résoudre les problèmes ;
4. exécuter `make` pour compiler ;

Le professeur

5. exécuter `sudo make install` pour installer le programme.



Mathieu Nebra

Entrepreneur à plein temps, auteur à plein temps et co-fondateur d'OpenClassrooms :)
Je pense vous de ce cours ?

Découvrez aussi ce cours en...

J'AI TERMINÉ CE CHAPITRE ET JE PASSE AU SUIVANT



Livre



PDF

AL... LE RÉSEAU ET FILTRER LE
 TRAFIC AVEC UN PARE-FEU

QUIZ : QUIZ 4



OPENCLASSROOMS

Qui sommes-nous ?

Financements

Expérience de formation

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

