

Formation GNU/Linux Debian : La ligne de commande

Dave Null (/dev/null)

22 Février 2016

Table des matières

1	Rappel des commandes	4
1.1	Introduction	4
1.1.1	Objectif du document	4
1.1.2	Conventions d'écriture	4
1.2	Navigation dans les répertoires	5
1.2.1	Contexte	5
1.2.2	Navigation dans les répertoires	6
1.2.3	Lister les fichiers et répertoires	7
1.2.4	Créer et modifier un fichier texte	8
1.2.5	Renommer/déplacer un fichier ou dossier	8
1.2.6	Supprimer un fichier ou dossier	9
1.3	Installations, suppressions et mises à jour	10
1.3.1	Contexte	10
1.3.2	Gestionnaire apt-get (installation depuis les dépôts)	11
1.3.3	Gestionnaire dpkg (installation à partir de .deb)	12
	Glossaire	14

Formation GNU/Linux Debian

1 Rappel des commandes

1.1 Introduction

1.1.1 Objectif du document

Ce document liste et explique des commandes utilisées lors de la formation au système d'exploitation **GNU/Linux**, et plus exactement la **distribution** Debian, découpés par partie selon le plan suivant :

- Navigation dans la fichiers et dossiers
 - Se déplacer dans les dossiers
 - Lister les fichiers et les répertoires
 - Créer et modifier des fichiers textes
 - Renommer et déplacer un fichier ou un répertoire
 - Supprimer un fichier ou un répertoire
- Installation, suppression, et mise à jour des logiciels
 - Installation, suppression et mises à jours avec **apt-get**
 - Installation, suppression avec **dpkg**

1.1.2 Conventions d'écriture

Les mots clés seront en **gras**, et il un **glossaire** est à la fin de ce document, tous les mots qui sont définis sont écrits en **rouge gras** dans le corps du texte, et contiennent un lien qui renvoi vers la bonne page du glossaire, où le terme correspondant est défini.

1.2 Navigation dans les répertoires

1.2.1 Contexte

Cette partie concerne les commandes pour naviguer dans les **répertoires**, c'est à dire se déplacer dans d'un « dossier » à l'autre, lister les fichiers, créer, et modifier des fichiers texte, ainsi que supprimer des fichiers ou les **répertoires**.

A noter que sous les systèmes **GNU/Linux** (et BSD)

- Les majuscules et minuscules sont différenciées dans les noms de fichier, de **répertoires**, ainsi que dans les commandes, **documents** et **Documents** ne sont pas le même **répertoire**
- Vous pouvez utiliser la touche tab pour compléter les commandes et les chemins vers des fichiers/**répertoires**, appuyer deux fois tab pour lister les possibilités en fonction de ceux que vous avez déjà commencé à taper
- Que ce soit dans une interface graphique ou en ligne de commande, les noms des **répertoire**/fichiers sont séparés pas des slash (/), comme dans une adresse web, et non pas des antislash (\) comme sous windows.
- Il n y pas d'espaces avant ou après les noms de fichiers/dossiers, ex :
/home/\$USER/Documents, et pas /home/\$USER/ Documents ou /home/\$USER /Documents.

NOTE : Dans les exemples suivants, des noms de fichiers et de dossiers sont donnés à titre d'exemple, il faut bien évidemment remplacer ces noms d'exemples par les vrais noms pour adapter les commandes.

De même, **/home/user** représente votre dossier **utilisateur**, il faut donc remplacer user par le nom correspondant. Sur les machines de l'association, ça donne **/home/\$USER**. Le **chemin complet** vers un fichier finit toujours par le nom de ce fichier, et non pas par le nom du dossier qui le contient (car il peut contenir plusieurs fichiers, donc il faut préciser sur quel fichier on cible sa commande). Si les noms de vos fichiers ou dossiers contiennent des espaces, il suffit d'encadrer le chemin entier (nom de fichier inclus) par des guillemets, par exemple pour déplacer un **fichier 1** du **répertoire courant** vers **/home/\$USER/dossier 2**, ça donnerait :

```
mv "/home/$USER/fichier 1" "/home/$USER/dossier 2"
```

1.2.2 Navigation dans les répertoires

Action	Commande	Commentaires
Aller vers le dossier spécifié, par exemple /home/user/Documents	cd /home/user/Documents	cd = change directory (changer de repertoire) Il faut remplacer user par votre nom d'utilisateur par ex /home/\$USER pour les PC de l'asso
Revenir au repertoire parent	cd ../	répertoire un niveau au dessus
Revenir 2 niveaux au dessus	cd ../../	On met autant de ../ que de dossiers qu'on veut remonter, sans droits administrateurs (root), on utilise que /home et ses sous-dossiers personnels (/home/nom_utilisateur)
Revenir 3 niveaux au dessus	cd ../../../	
Revenir au dossier /home/user	cd	Cette commande mène directement vers le dossier utilisateur sans avoir à donner le chemin
Savoir dans quel repertoire on se trouve	pwd	pwd = print working directory , pwd affiche le chemin complet vers le répertoire courant

1.2.3 Lister les fichiers et répertoires

Action	Commande	Commentaires
Lister les fichiers et répertoires du répertoire courant	ls	ls = list Affiche les fichiers dossiers du répertoire courant mais pas celui des sous-répertoires
Lister les fichiers et répertoires du répertoire courant en incluant aussi ceux de chaque sous-répertoire, ainsi que ceux des sous-répertoires jusqu'au dernier niveau	ls -R	Le résultat de cette commande peut être très long (et lent) si vous avez beaucoup de dossiers, mais c'est possible de cibler vers des sous dossier. A ne pas confondre avec ls -r qui fait la même chose que la commande précédente, en inversant l'ordre d'affichage (de Z à A)
Lister les fichiers et répertoires dans un autre repertoire, par exemple /home	ls /home	Peut-être combiné avec la commande précédente, par exemple ls -R /chemin/vers/ce/dossier pour avoir un résultat plus lisible

1.2.4 Créer et modifier un fichier texte

Action	Commande	Commentaires
Créer un fichier <code>test.txt</code> dans le répertoire courant	<code>touch test.txt</code>	C'est aussi possible d'indiquer un chemin complet vers le fichier, s'il ne se trouve pas dans le répertoire courant : <code>touch /home/user/fichier.txt</code>
Modifier un fichier <code>test.txt</code>	<code>nano test.txt</code>	Comme d'habitude, on peut soit mettre que le nom du fichier s'il est dans le répertoire courant sinon donner le chemin vers ce fichier

1.2.5 Renommer/déplacer un fichier ou dossier

Action	Commande	Commentaires
Renommer « <code>anciennom</code> » en « <code>nouveaunom</code> »	<code>mv anciennom nouveaunom</code>	<code>mv = move</code> (déplacer) car renommer un fichier consiste en réalité à le déplacer, en le remettant au même endroit mais sous un autre nom
Déplacer un fichier <code>test</code> depuis le répertoire courant vers le dossier <code>/home/user/DirTest</code>	<code>mv test /home/user/DirTest</code>	Il s'agit bien de la même commande que pour renommer le fichier, <code>mv</code> , sauf que pour déplacer le fichier, on met, à la place du nouveau nom, le chemin vers le dossier dans le quel on souhaite déplacer le fichier

1.2.6 Supprimer un fichier ou dossier

Action	Commande	Commentaires
Supprimer « fichier1 »	rm fichier1	rm = remove (supprimer) Comme d'habitude, on indique le nom du fichier à supprimer, s'il est dans le répertoire courant , sinon on indique tout le chemin, qui fini par le nom du fichier à supprimer
Supprimer répertoire « rep1 » plein, ainsi que tout son contenu	rm -r rep1	L'option -r (recursif) indique qu'il faut supprimer tout ce qu'il y a dans le dossier ciblé (ici rep2), en incluant les fichiers et dossiers qui s'y trouvent, ainsi que leurs fichiers et sous-dossiers, jusqu'au dernier sous-dossier
Supprimer répertoire « rep2 » vide et seulement s'il est vide	rmdir rep2	rmdir = remove directory (supprimer le répertoire) Si le dossier contient des fichiers ou sous-dossiers , rmdir renvoie une erreur , tout comme rm sans l'option -r pointé vers un dossier plein, ce qui permet d'éviter de supprimer par erreur la totalité d'un répertoire et son contenu, si on veut juste supprimer un fichier dont on oublie de rajouter le nom en fin de commande

1.3 Installations, suppressions et mises à jour

1.3.1 Contexte

Cette partie concerne l'utilisation d'un **gestionnaire de paquets** pour installer, désinstaller ou mettre à jour un **paquet** ainsi que ses **dépendance**, c'est dire les différents logiciels que vous utiliser, mais aussi ce dont le système à besoin de fonctionner.

Un **gestionnaire de paquets** est un moyen particulièrement efficace pour gérer les mises de l'ensemble de son système, de façon centralisée, c'est à dire sans avoir à chercher les différents logiciels à gauche à droite. Le **gestionnaire de paquets** gère aussi les **dépendances**.

C'est aussi un moyen de récupérer ses logiciels d'une source relativement sûre (même si la sécurité absolue n'existe pas en informatique) sans risquer de se retrouver avec des versions infectées des logiciels (par exemple des logiciels qui affichent des pubs intempestives, changent la page d'accueil du navigateur web, et y installent des barres d'outils de provenance douteuse).

Pour installer un logiciel, il faut être **administrateur**, sous les systèmes basés sur un noyau **Linux** (de même pour BSD), l'administrateur est dit **root** (racine) car il a accès à tout le système depuis la "racine", qui correspond au "dossier" /, / contient, vous l'aurez deviné, tout le système.

Pour obtenir les droits root depuis un terminal, il suffit de taper la commande **su** (abréviation de super-user parce que c'est un compte utilisateur qui tous les droits sur le système) et de taper le mot de passe root, qui ne sera pas visible sur le terminal lors que vous tapez (même pas sous forme de caractères astérisque [*] ou de bulle [•], pour des raisons de sécurité), mais il sera bien pris en compte.

Par souci de simplification, ce document ne donne pas les instructions pour ajouter des dépôts tierces (ainsi que les vérifications qui en découlent), ni les options les plus avancées des gestionnaires du paquets. Mais pour les plus curieux, vous pouvez directement le mode d'emploi (inclus dans le système) des **gestionnaire de paquets apt-get** et de **dpkg**, accessibles respectivement avec les commandes **man apt-get** et **man dpkg**, la commande **man**, qui signifie manual (manuel d'instructions), existe pour la grande majorité des outils en ligne de commande sous **GNU/Linux**, et donne automatiquement accès à la version française du mode d'emploi (si elle existe pour un outil donné) car le système à été configuré en français.

1.3.2 Gestionnaire apt-get (installation depuis les dépôts)

Cette partie concerne l'utilisation gestionnaire de paquets **apt-get** pour récupérer un logiciel depuis les dépôts officiels, ou le supprimer, ainsi que pour mettre à jour son système, toujours depuis les dépôt.

Action	Commande	Commentaires
Vérifier le contenu des dépôts	apt-get update	Avant de mettre à jour son système, ou installer un logiciel un logiciel, il faut mettre à jour les dépôts pour avoir accès à dernières mises à jours
Installer un logiciel, par exemple gimp	apt-get install gimp	Pour n'importe quel autre logiciel présent dans les dépôts, il suffit de remplacer gimp par le nom du logiciel voulu
Désinstaller un logiciel, par exemple gimp	apt-get remove gimp	remove = supprimer Pour n'importe quel autre logiciel installé, il suffit de remplacer gimp par le nom du logiciel qu'on veut supprimer
Mettre à jour l'ensemble de son système	apt-get upgrade	Note : Ne pas confondre update qui fait la mise à jour des dépôts , avec upgrade qui met à jour le système complet, il faut toujours faire d'abord update , puis upgrade

1.3.3 Gestionnaire dpkg (installation à partir de .deb)

Contrairement à **apt-get**, le gestionnaire de paquets **dpkg** sert à installer des logiciels depuis un fichier en **.deb** préalablement récupéré sur Internet. Dans de rares cas, il se peut qu'un logiciel ne soit pas présent dans les dépôts officiels, ni dans des dépôts tierces, et qu'il nécessite peu ou pas de **dépendances**, et qu'il soit distribué sous forme de point **.deb**. Et dans le cas où les **dépendances** ne sont pas préinstallées mais qu'elles sont disponibles, **dpkg** indique une commande **apt-get** à utiliser, pour installer les dépendances manquantes. Si les **dépendances** ne sont ni préinstallées, ni dans les dépôts, elles peuvent aussi être distribuées dans fichiers **.deb** avec le logiciel qui en a besoin pour fonctionner (mais c'est un cas extrêmement rare).

Action	Commande	Commentaires
Installer fichier.deb	dpkg -i fichier.deb	Option -i = install Se rendre dans le dossier où se trouve fichier.deb (sinon on peut aussi donner le chemin complet vers fichier.deb) Le nom « fichier.deb » est à remplacer bien sûr avec le nom du .deb que vous voulez installer. La version installée dépendra du .deb, généralement le numéro de version est indiquée dans le nom du .deb, par exemple <code>gimp_2.8_amd64.deb</code> , donc il faut prendre la dernière version au moment de télécharger le fichier, généralement les développeurs du logiciel s'arrangent pour mettre en évidence la dernière version sur leur site
Désinstaller un logiciel, par exemple gimp	dpkg -r gimp	Option -r = remove Contrairement à l'installation avec dpkg , on ne met pas fichier.deb pour la suppression, mais juste le nom du paquet à supprimer, sans le .deb (ni le nom complet du fichier) Pour n'importe quel autre logiciel installé, il suffit de remplacer <code>gimp</code> par le nom du logiciel qu'on veut supprimer

Glossaire

Dépendance Ce terme désigne des "morceaux" de logiciels (aussi nommées "bibliothèque") **nécessaires au fonctionnement** d'un ou plusieurs autre(s) logiciel(s) Elle est, dans l'extrême majorité des cas, installée (et supprimée, si elle n'est plus utile) **automatiquement** par le **gestionnaire de paquets** en fonction des logiciels qu'on lui demande d'installer ou de supprimer. Une dépendance est très souvent **partagée** entre plusieurs logiciels, au lieu d'accumuler les mêmes bibliothèques plusieurs fois (une fois pour chaque logiciel qui s'en sert), d'où la nécessité de passer par un **gestionnaire de paquets** qui s'occupe de tout gérer. Cette approche (partage des dépendances) permet d'avoir un système relativement compact et rapide.. 10, 12

Dépôt Serveur dédié à une **Distribution**, contenant des paquets. 11

Distribution Ce terme désigne un système d'exploitation de type **Linux** ou **BSD**, **entier** (c'est à dire avec des **pilotes** inclus) et **prêt à l'emploi** (un ensemble de logiciels y sont préinstallés), par exemple Debian. Chaque distribution a ses **particularités** et réponds à un certains cas d'usages (bureautique, multimedia, serveurs. . . . 4

Gestionnaire de paquets Logiciel servant gérer son système en facilitant les **installations, suppressions** et **mise à jour** des **paquets**, par exemple **apt-get** pour **Debian** et **Linux Mint**, ou **yum** pour **Fedora**. Cet outil permet généralement aussi de rajouter ou des retirer des dépôts tierces permettant d'avoir accès à des logiciels manquants dans les dépôts de base (ce qui est rare et ne concerne que des cas particuliers). 10–12

GNU Ensemble de logiciels libres qu'on ajoute habituellement au noyau **Linux** pour en faire un système complet, utilisable par l'humain. 4

Linux Nom donné aux **distributions** basés sur le noyau libre Linux. Ce terme souvent utilisé comme abréviation de **GNU/Linux** pour désigner tous les systèmes qui utilisent aussi le projet **GNU**, en plus de **Linux** (ex : Debian, Ubuntu, Mint, Fedora. . .) mais plus rarement pour parler des systèmes Linux sans la partie GNU (ex : Android, Firefox OS). 4

Paquet Un logiciel ou une **dépendance**, qu'on installe, on supprime, et on met à jour en passant par un gestionnaire de paquets. 10, 13

Recurif Une action qui se répète elle même. 9

Répertoire Un répertoire un type de fichiers qui contient d'autres fichiers et répertoires, on utilise souvent le terme "dossier" dans les systèmes windows. 4, 5, 9

Répertoire courant Le terme répertoire désigne le repertoire dans lequel on se situe à l'instant t. 5-9