

## Reprenez le contrôle à l'aide de Linux !

🕒 30 heures 📶 Facile

Mis à jour le 29/06/2021



### RTFM : lisez le manuel !

Quand on vient de Windows, on n'a pas trop l'habitude de lire des documentations. Parfois les logiciels sont livrés avec des modes d'emploi, mais honnêtement, qui ici prend la peine de les lire ?

Sous Linux, lire la documentation doit devenir un **réflexe**. En effet, bien que cela fasse un peu peur au premier abord, la documentation est vraiment le meilleur endroit pour en savoir plus sur les commandes que vous utilisez.

Les livres que j'écris ne pourront jamais rivaliser avec la documentation. Je peux vous montrer les commandes et les paramètres qui me semblent les plus utiles, mais pour connaître certains paramètres dont vous avez besoin moins souvent, vous n'y couperez pas : **vous aurez besoin de lire la doc**.

Ce chapitre est justement là pour « démystifier » le manuel et vous apprendre à le lire. C'est peut-être un des chapitres les plus importants du cours, car si vous savez lire la doc, vous êtes capables d'apprendre tout ce dont vous avez besoin... et vous pourrez donc tout faire !

Le titre de ce chapitre est une insulte « amicale » entre linuxiens. C'est en général ce qu'on dit à un débutant qui pose trop de questions sur les forums alors que la réponse se trouve dans la doc.  
« RTFM » est une abréviation qui signifie « *Read The Fucking Manual* », ce qui se traduirait en français par quelque chose comme « Lis ce p\*\*\* de manuel ». Mais oui, c'est amical ! Puisque je vous le dis...

### man : afficher le manuel d'une commande

La commande magique que nous allons expérimenter tout au long de ce chapitre est `man`, qui est l'abréviation de **manual**.

La commande `man` s'utilise très simplement : elle prend en paramètre le nom de la commande dont vous voulez lire la doc.

Par exemple, si je veux tout savoir sur la commande `mkdir`, je dois écrire :

```
man mkdir
```

Votre console devrait alors ressembler à l'image de la figure suivante.

```
Terminal - mateo21@simple-it-pc1: ~
Bichier Éditer Affichage Terminal Aller Aide
MKDIR(1) User Commands MKDIR(1)
NAME
mkdir - make directories
SYNOPSIS
mkdir [OPTION] DIRECTORY...
DESCRIPTION
Create the DIRECTORY(ies), if they do not already exist.

Mandatory arguments to long options are mandatory for short options too.

-Z, --context=CONTEXT (SELinux) set security context to CONTEXT
-m, --mode=MODE
    set permission mode (as in chmod), not rwxrwxrwx - umask
-p, --parents
    no error if existing, make parent directories as needed
-v, --verbose
Manual page mkdir(1) line 1
```

Le manuel de la commande mkdir

Il s'agit de la documentation de la commande `mkdir`. Là-dedans, il y a tout ce qu'il faut savoir sur `mkdir`.

La doc de la commande étant généralement un peu longue, celle-ci s'affiche page par page, à la manière de `less` qu'on a vue dans un chapitre précédent.

### Se déplacer dans le manuel

Voici quelques commandes à connaître pour se déplacer dans le manuel.

- Utilisez les **touches fléchées** du clavier (vers le haut et vers le bas) pour vous déplacer ligne par ligne.
- Vous pouvez utiliser les touches `Page Up` et `Page Down` (ou `Espace`) pour vous déplacer de page en page.
- Appuyez sur la touche `Home` (aussi appelée `Origine`) pour revenir au début du manuel, et sur `Fin` pour aller à la fin.

- Appuyez sur la touche `/` (slash) pour effectuer une recherche ; c'est très pratique ! Tapez ensuite le mot que vous recherchez dans le manuel puis appuyez sur `Entrée` . Si la recherche renvoie un résultat vous serez automatiquement placés sur le premier résultat trouvé. Pour passer au résultat suivant, tapez à nouveau `/` puis directement sur `Entrée` (sans retaper votre recherche).
- Appuyez sur la touche `Q` pour quitter le manuel à tout moment, comme vous le faisiez avec `less` .



Si comme moi vous avez ouvert votre manuel dans une console graphique, vous pouvez aussi utiliser la molette de la souris !

## Les principales sections du manuel

Comme vous pouvez le voir, le manuel de la commande est découpé en plusieurs sections (leurs noms sont écrits en gras et alignés à gauche de l'écran).

Voici leur signification.

- **NAME** : le nom de la commande dont vous êtes en train d'afficher le manuel ainsi qu'une courte description de son utilité.
- **SYNOPSIS** : c'est la liste de toutes les façons d'utiliser la commande. Nous y reviendrons un peu plus loin car il est vital de comprendre ce qui est écrit dans cette section.
- **DESCRIPTION** : une description plus approfondie de ce que fait la commande. On y trouve aussi la liste des paramètres et leur signification. C'est en général la section la plus longue.
- **AUTHOR** : l'auteur du programme. Il y a parfois de nombreux auteurs ; c'est souvent le cas d'ailleurs avec le logiciel libre.
- **REPORTING BUGS** : si vous rencontrez un bug dans le logiciel, on vous donne l'adresse de la personne à contacter pour le rapporter.
- **COPYRIGHT** : le *copyright*, c'est-à-dire la licence d'utilisation de la commande. La plupart des programmes que vous utilisez sont certainement des programmes open source sous licence GPL, ce qui vous donne le droit de voir la source et de redistribuer le programme librement.
- **SEE ALSO** : cette section vous propose de « voir aussi » d'autres commandes en rapport avec celle que vous êtes en train de regarder. C'est une section parfois intéressante.

Pour faire simple, les trois premières sections sont vraiment les plus importantes ( **NAME** , **SYNOPSIS** et **DESCRIPTION** ). Ce sont celles que nous regarderons dans la plupart des cas et c'est aussi pour cela qu'elles sont au début du manuel. :-)

## La langue des pages du manuel



Eh mais... c'est tout en anglais ! Tout le monde n'est pas un crack en anglais... comment on fait, nous, si on a du mal à lire l'anglais ?

Vous n'allez peut-être pas apprécier, mais je tiens à vous donner un conseil, le meilleur : familiarisez-vous avec l'anglais. Si vous voulez travailler dans l'informatique, c'est de toute façon une langue incontournable ; il est impossible de l'ignorer ou ce serait alors du pur suicide.

Bon. Malgré tout, il y en a peut-être parmi vous qui ne comptent pas forcément travailler dans l'informatique mais qui aimeraient éviter d'avoir à apprendre l'anglais pour se servir de Linux, ce que je peux très bien comprendre. Si l'évocation du mot « anglais » provoque chez vous des éruptions cutanées inexplicables, alors voici une bonne nouvelle : il existe une version française des pages de manuel !

Vous pouvez l'installer grâce à la commande `apt-get` qu'on a justement apprise dans le chapitre précédent comme par hasard (rien n'est laissé au hasard, je vous l'ai dit) :

```
apt-get install manpages-fr
```

Le paquet à installer est `manpages-fr` comme vous pouvez le voir.

Maintenant, si vous tapez `man mkdir` , vous verrez la page s'afficher en français (figure suivante).

```
Terminal - mateo21@simple-it-pci: ~
Bichier Éditer Affichage Terminal Aller Aide
MKDIR(1) Commandes MKDIR(1)

NOM
  mkdir - Créer des répertoires

SYNOPSIS
  mkdir [OPTION] RÉPERTOIRE...

DESCRIPTION
  Créer les RÉPERTOIRES s'il n'existent pas.

  -Z, --context=CONTEXTE (SELinux) imposer CONTEXTE comme contexte de
    sécurité

  Les paramètres obligatoires pour les options de forme longue le sont
  aussi pour les options de forme courte.

  -m, --mode=MODE
    définir MODE comme droit d'accès (comme dans chmod), mais pas au
    format umask (rwxrwxrwx)

  -p, --parents
    créer des répertoires parent (répertoires intermédiaires) si

Manual page mkdir(1) ligne 1
```

Pages du manuel en français



Utiliser les pages du manuel en français **n'est pas recommandé**. En effet, ces pages ne sont parfois pas à jour, certaines ne sont pas traduites, et seule la version anglaise est assurée d'être la plus à jour et de comporter le moins d'erreurs. D'ailleurs, en parlant d'erreur cher lecteur, une faute d'orthographe s'est glissée dans la capture d'écran de la figure ci-dessus, sauras-tu la retrouver ?

Dans la suite de ce livre, je considérerai que vous travaillez avec les pages du manuel **en anglais**. C'est vraiment ce que je recommande. Utilisez la version française uniquement si vous ne comprenez rien à l'anglais, sinon faites un petit effort ; je vous assure que ça vaut le coup.

Je vais donc désinstaller les pages du manuel en français et revenir à la version anglaise pour la suite de ce chapitre :

```
apt-get autoremove manpages-fr
```

Bien, les présentations étant faites, passons à la suite !

Nous allons apprendre à lire la section `SYNOPSIS`, une des sections les plus importantes.

## Comprendre le SYNOPSIS

Le `SYNOPSIS` est une des sections les plus importantes mais aussi une des plus difficiles à lire.



Quel est le rôle du `SYNOPSIS` ?

Son rôle est de **lister toutes les façons possibles d'utiliser la commande**. En clair, le `SYNOPSIS` vous affiche toutes les combinaisons de paramètres que l'on peut réaliser avec cette commande.

Certains `SYNOPSIS` sont simples, d'autres plus compliqués. Je pense que le mieux est de voir des exemples pour bien comprendre comment ça fonctionne.

```
man mkdir
```

Commençons par le man de `mkdir`. La section d'introduction du manuel nous dit « `mkdir - make directories` », ce qui signifie que `mkdir` sert à créer des répertoires.

```
SYNOPSIS
```

La section `SYNOPSIS` de `mkdir` est présentée sur la figure suivante.

```
mkdir [OPTION] DIRECTORY...
```

SYNOPSIS de mkdir

Même si ce `SYNOPSIS` est court, il contient déjà beaucoup d'informations : il vous dit comment on doit utiliser la commande.

Détaillons point par point ce `SYNOPSIS`.

- `mkdir` : pour utiliser la commande `mkdir`, vous devez commencer par taper `mkdir` ; ça, c'est logique.
- `[OPTION]` : après `mkdir`, vous pouvez écrire une option. Dans le `SYNOPSIS`, on met des crochets pour indiquer que c'est facultatif. Vous n'êtes donc pas obligés d'écrire une option.
- `DIRECTORY` : c'est le nom du répertoire à créer. Ce paramètre est obligatoire puisqu'il n'est pas entre crochets. C'est en effet logique : la commande `mkdir` sert à créer un dossier, la moindre des choses est d'indiquer le nom du dossier à créer !
- `...` : le terme `DIRECTORY` est suivi de points de suspension. Cela signifie que l'on peut répéter `DIRECTORY` autant de fois que l'on veut. Traduction : on peut indiquer plusieurs répertoires à la fois pour que la commande les crée tous d'un coup.



Pourquoi `mkdir` est écrit en gras tandis que `OPTION` et `DIRECTORY` sont soulignés ?

Les mots du `SYNOPSIS` écrits en gras sont des mots à taper tels quels. Les mots soulignés, eux, doivent être remplacés par le nom approprié.

C'est logique : on doit bel et bien écrire précisément `mkdir`, par contre on ne doit pas écrire `DIRECTORY` mais le nom du répertoire. La présence d'un paramètre souligné signifie donc : « Remplacez le mot souligné par un mot qui convient à votre cas ».

### Exemples d'utilisation

D'après le `SYNOPSIS`, on doit au minimum écrire le nom du dossier. Par exemple :

```
mkdir images
```

Comme on l'a vu, les points de suspension après `DIRECTORY` nous indiquent qu'on peut répéter le nom du répertoire autant de fois que l'on veut, ce qui nous permet d'en créer plusieurs d'un coup.

On peut donc aussi utiliser `mkdir` comme ceci :

```
mkdir images videos musiques
```

... ce qui aura pour effet de créer trois dossiers : `images`, `videos` et `musiques`.

Maintenant, on peut aussi préciser des options facultatives. Ces options sont listées dans la section `DESCRIPTION` du man juste un peu plus bas :

```
DESCRIPTION
```

```
Create the DIRECTORY(ies), if they do not already exist.
```

```
Mandatory arguments to long options are mandatory for short options too.
```

```
-m, --mode=MODE
    set file mode (as in chmod), not a=rwx - umask
```

```
-p, --parents
    no error if existing, make parent directories as needed
```

```
-v, --verbose
    print a message for each created directory
```

```
-Z, --context=CTX
    set the SELinux security context of each created directory to
```

CTX

```
--help display this help and exit

--version
    output version information and exit
```

Toutes ces options peuvent être utilisées à la place de `[OPTION]` dans le `SYNOPSIS`.

Par exemple, l'option `-v` (ou `--verbose`), c'est pareil mais plus long) affiche un message après chaque répertoire créé. On peut donc écrire :

```
mkdir -v images videos musiques
```

Résultat :

```
mateo21@mateo21-desktop:~/tests$ mkdir -v images videos musiques
mkdir: création du répertoire `images'
mkdir: création du répertoire `videos'
mkdir: création du répertoire `musiques'
```

La commande nous informe maintenant de ce qu'elle fait. Sans le `-v`, la commande n'affiche rien (on dit qu'elle est silencieuse).

Vous remarquerez d'ailleurs qu'on retrouve l'option `-v` dans beaucoup de commandes. Elle a chaque fois la même signification : elle demande à la commande d'afficher le détail de ce qu'elle est en train de faire. On dit alors qu'on utilise la commande en mode « verbeux » (bavard), pour bien voir tout ce qu'elle fait.

`man cp`

Essayons une commande un peu plus complexe : `cp`. Je vous rappelle que cette commande sert à copier des fichiers et des répertoires.

### SYNOPSIS

Son `SYNOPSIS` est présenté sur la figure suivante.

```
cp [OPTION]... [-T] SOURCE DEST
cp [OPTION]... SOURCE... DIRECTORY
cp [OPTION]... -t DIRECTORY SOURCE...
```

SYNOPSIS de cp

Là, ça commence à devenir un peu plus délicat.

Pourquoi y a-t-il trois lignes, déjà ? Tout simplement parce qu'on ne pouvait pas exprimer toutes les façons d'utiliser `cp` sur une seule ligne.

- Commençons par la première ligne :

`cp [OPTION]... [-T] SOURCE DEST` La seule chose obligatoire ici ce sont les paramètres `SOURCE` (le nom du fichier à copier) et `DEST` (le nom de la copie à créer). Ces fichiers peuvent être précédés d'une ou plusieurs options (remarquez les points de suspension) ainsi que de l'option `-T`.

- La seconde ligne est un peu différente :

`cp [OPTION]... SOURCE... DIRECTORY` Cette fois, on peut copier un ou plusieurs fichiers (`SOURCE...`) vers un répertoire (`DIRECTORY`). Tout cela peut encore une fois être précédé d'une ou plusieurs options.

- Enfin, la troisième ligne :

`cp [OPTION]... -t DIRECTORY SOURCE...` ... signifie qu'on peut aussi écrire le répertoire (`DIRECTORY`) dans un premier temps, suivi d'un ou plusieurs fichiers (`SOURCE...`). Attention, vous remarquez que dans ce cas il est obligatoire d'utiliser le paramètre `-t` qui n'est plus entre crochets.

### Exemples d'utilisation

Ça fait beaucoup de façons d'utiliser `cp`, en fait.

Si on se base sur la première ligne, on peut juste écrire :

```
cp photo.jpg photo_copie.jpg
```

... ce qui aura pour effet de créer la copie `photo_copie.jpg`.

On peut aussi ajouter une ou plusieurs options. Pour connaître toutes les options disponibles, vous devrez lire la section `DESCRIPTION`. Pour `cp`, il y a beaucoup de choix comme vous pouvez le voir.

Par exemple, on retrouve notre mode `-v` (verbeux) qui demande à la commande de détailler ce qu'elle fait. On pourrait aussi ajouter `-i` qui demande confirmation si le fichier de destination existe déjà.

On peut donc faire :

```
cp -vi photo.jpg photo_copie.jpg
```

Dans mon cas, le fichier `photo_copie.jpg` existait déjà. L'ajout de l'option `-i` va me demander confirmation pour savoir si je veux bien écraser le fichier. Je peux répondre par « o » ou « n » (pour oui ou non), ou « y » ou « n » (pour yes ou no).

```
mateo21@mateo21-desktop:~$ cp -vi photo.jpg photo_copie.jpg
cp: écraser `photo_copie.jpg'? o
`photo.jpg' -> `photo_copie.jpg'
```

Comme le fichier existait déjà, on m'a demandé confirmation. La dernière ligne est le résultat du mode verbeux qu'on a demandé.

Bien. Tout ça c'était juste pour la première ligne, dans le cas où l'on veut copier un fichier. Essayons un peu ce que propose la seconde ligne : copier un ou plusieurs fichiers dans un dossier.

```
cp photo.jpg photo_copie.jpg images/
```

Là, on exploite la seconde façon d'utiliser `cp` (seconde ligne du `SYNOPSIS`). On copie deux fichiers dans le sous-dossier `images/`.

Bien entendu, comme l'indique le `SYNOPSIS`, on peut là encore utiliser des options, comme `-v` et `-i` que l'on vient de voir.

**i** Mon conseil : vous vous demandez peut-être comment je comprends la signification de chaque mot du `SYNOPSIS`. Par exemple, était-il évident de savoir que `SOURCE` correspondait au fichier que l'on voulait copier ? Si je sais tout ça, c'est parce que j'ai lu attentivement le début de la section `DESCRIPTION` du man. Je vous recommande d'en faire de même quelle que soit la commande que vous êtes en train d'analyser.

#### DESCRIPTION

Copy SOURCE to DEST, or multiple SOURCE(s) to DIRECTORY.

**i** Cette phrase est claire, pour peu que l'on comprenne l'anglais. Elle signifie que l'on peut copier le paramètre appelé `SOURCE` vers `DEST`, ou encore (cas de la seconde ligne) plusieurs fichiers `SOURCE` vers un répertoire (`DIRECTORY`).

#### man apt-get

Plus joyeux encore : le `SYNOPSIS` de la commande `apt-get` (figure suivante).

```
apt-get [-hvs] [-o=option de configuration] [-c=fichier] {[update] |
[upgrade] | [dselect-upgrade] | [install paquet...] |
[remove paquet...] | [source paquet...] | [build-dep paquet...]}
| [check] | [clean] | [autoclean]}
```

SYNOPSIS de apt-get

La bonne nouvelle cette fois c'est que les créateurs d' `apt-get` n'ont pas eu besoin de faire tenir la commande sur trois lignes : ils ont mis tous les cas d'utilisation possibles sur une seule ligne !

La mauvaise nouvelle, c'est que c'est un peu difficile à lire comme ça.

Décortiquons.

La commande `apt-get` doit commencer par « apt-get » (ce mot est d'ailleurs écrit en gras). Ça, c'est logique.

Ensuite, vous pouvez utiliser une des options `-hvs` (vous pouvez utiliser juste `-h`, mais aussi `-hv`, `-v`, `-vs`, `-hvs` ...). Ces options étant entre crochets, elles sont facultatives.

Pareil ensuite pour `-o` et `-c` ; ces options sont facultatives. En revanche, vous remarquerez qu'elles doivent être obligatoirement suivies d'une valeur. Par exemple `-o=option de configuration`. Je vous rappelle que le fait que `option de configuration` soit souligné signifie que vous ne devez pas recopier ces mots tels quels dans la console : vous devez les remplacer par une valeur qui convient (lisez la section `DESCRIPTION` pour en savoir plus sur `-o`).

La section qui m'intéresse et que je voudrais qu'on analyse plus en détail arrive juste après. Elle commence et se termine par des accolades :

```
{[update] | [upgrade] | [dselect-upgrade] | [install paquet...] | [remove paquet...] | [source paquet...] | [build-dep paquet...] | [check] | [clean] | [autoclean]}
```

Vous remarquerez qu'à l'intérieur les mots sont séparés par des barres verticales « `|` ». Ces barres verticales signifient « OU », ce qui veut dire que vous devez mettre une et une seule option issue de la liste entre accolades.

Parmi ces options possibles, il y en a que vous devez connaître maintenant, comme :

- `update` : met à jour le cache des paquets disponibles sur votre ordinateur ;
- `upgrade` : met à jour tous les paquets installés si une nouvelle version est disponible ;
- `install paquet...` : installe le ou les paquets demandés. La présence des points de suspension après « paquet » signifie que vous pouvez indiquer plusieurs paquets à installer d'un coup ;
- etc.

Il y a bien d'autres mots clés utilisables. Pour voir la signification de chacun d'eux, je vous invite à lire la section `DESCRIPTION` du man qui sert précisément à expliquer cela.

#### Exemples d'utilisation

Le `SYNOPSIS` indique donc qu'on doit choisir une des options entre accolades séparées par des barres verticales.

On peut donc écrire :

```
apt-get install monpaquet
```

Ou encore :

```
apt-get update
```

Ou encore :

```
apt-get autoclean
```

En revanche, on ne peut pas utiliser simultanément deux options séparées par une barre verticale :

INTERDIT :  
apt-get update install monpaquet

Le **SYNOPSIS** nous avait bien dit : « Utilisez **update** OU **install** OU **upgrade** (OU ... ), mais pas deux éléments de cette liste à la fois ». Souvenez-vous donc que les barres verticales signifient « OU » et tout ira bien.

## Résumé de la syntaxe du SYNOPSIS

Voici un petit résumé de la syntaxe du **SYNOPSIS** pour vous souvenir de la façon dont chaque élément doit être interprété :

- **gras** : tapez le mot exactement comme indiqué ;
- **souligne** : remplacez le mot souligné par la valeur qui convient dans votre cas ;
- **[-hvc]** : toutes les options **-h**, **-v** et **-c** sont facultatives ;
- **a|b** : vous pouvez écrire l'option « a » OU « b », mais pas les deux à la fois ;
- **option...** : les points de suspension indiquent que l'option peut être répétée autant de fois que vous voulez.

## apropos : trouver une commande

Le man suppose que vous connaissez déjà votre commande et que vous voulez en savoir plus. Mais si vous ne connaissez pas la commande, comment faites-vous ?

C'est là que la commande **apropos** intervient. Vous lui donnez en paramètre un mot clé et elle va le rechercher dans les descriptions de toutes les pages du manuel.

La commande **apropos** est donc un peu l'inverse de **man** : elle vous permet de retrouver une commande.

Prenons un exemple : vous recherchez une commande (que vous avez installée) en rapport avec le son parce que vous aimeriez bien savoir comment modifier le volume en console. Vous pouvez taper :

```
apropos sound
```

... ce qui va rechercher toutes les commandes qui parlent de son ( **sound** ) dans leur page du manuel.

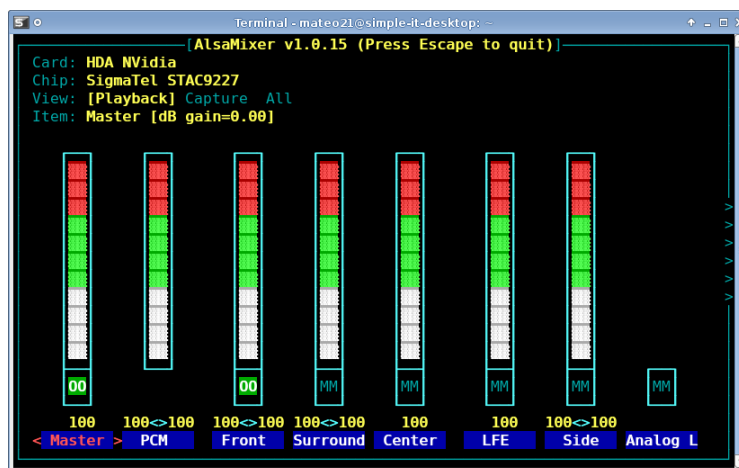
Résultat :

```
mateo21@mateo21-desktop:~$ apropos sound
alsactl (1)      - advanced controls for ALSA soundcard driver
alsamixer (1)   - soundcard mixer for ALSA soundcard driver, with ncurses...
amixer (1)      - command-line mixer for ALSA soundcard driver
aplay (1)       - command-line sound recorder and player for ALSA soundc...
arecord (1)     - command-line sound recorder and player for ALSA soundc...
artscat (1)     - pipe data to sound device
asoundconf (1)  - utility to read and change the user's ALSA library con...
```

À gauche la commande, à droite l'extrait de sa courte description dans laquelle **apropos** a trouvé le mot que vous recherchiez.

Il se trouve que ce que je cherchais était **alsamixer** (figure suivante). Et zou !:-)

```
alsamixer
```



Et si vous voulez en savoir plus sur la commande **alsamixer**, vous savez maintenant comment faire !

```
man alsamixer
```

```

Terminal - mateo21@simple-it-desktop:
ALSAMIXER(1)
NAME
    alsamixer - soundcard mixer for ALSA soundcard driver, with ncurses
    interface
SYNOPSIS
    alsamixer [options]
DESCRIPTION
    alsamixer is an ncurses mixer program for use with the ALSA soundcard
    drivers. It supports multiple soundcards with multiple devices.
OPTIONS
    -h, -help
        Help: show available flags.
    -c <card number or identification>
        Select the soundcard to use, if you have more than one. Cards
        are numbered from 0 (the default).
    -D <device identification>
        Select the mixer device to control.
Manual page alsamixer(1) line 1

```

Son `SYNOPSIS`, présenté sur la figure suivante, est ridiculement simple. Pfeuh ! Même pas drôle.

## D'autres façons de lire le manuel

Bien que ce soit la technique la plus courante, utiliser `man` et `apropos` n'est pas le seul moyen de vous documenter.

Quelles sont les alternatives à `man` ?

### Le paramètre `-h` (et `--help`)

Bien que ça ne soit pas une règle, la plupart des commandes acceptent un paramètre `-h` (et parfois son équivalent plus long `--help`) qui provoque l'affichage d'une aide résumée. Parfois cette aide est d'ailleurs plus facile à lire que celle du `man`, ce qui fait qu'il m'arrive de l'utiliser de temps en temps.

Par exemple :

```
apt-get -h
```

Ce qui nous donne :

```

mateo21@mateo21-desktop:~$ apt-get -h
apt 0.7.9ubuntu15 pour amd64 compilé sur Mar 14 2008 00:00:28
Usage: apt-get [options] command
    apt-get [options] install|remove pkg1 [pkg2 ...]
    apt-get [options] source pkg1 [pkg2 ...]

apt-get is a simple command line interface for downloading and
installing packages. The most frequently used commands are update
and install.

Commands:
  update - Retrieve new lists of packages
  upgrade - Perform an upgrade
  install - Install new packages (pkg is libc6 not libc6.deb)
  remove - Remove packages
  autoremove - Remove all automatic unused packages
  purge - Remove and purge packages
  source - Download source archives
  build-dep - Configure build-dependencies for source packages
  dist-upgrade - Distribution upgrade, see apt-get(8)
  dselect-upgrade - Follow dselect selections
  clean - Erase downloaded archive files
  autoclean - Erase old downloaded archive files
  check - Verify that there are no broken dependencies

Options:
  -h This help text.
  -q Loggable output - no progress indicator
  -qq No output except for errors
  -d Download only - do NOT install or unpack archives
  -s No-act. Perform ordering simulation
  -y Assume Yes to all queries and do not prompt
  -f Attempt to correct a system with broken dependencies in place
  -m Attempt to continue if archives are unlocatable
  -u Show a list of upgraded packages as well
  -b Build the source package after fetching it
  -V Show verbose version numbers
  -c=? Read this configuration file
  -o=? Set an arbitrary configuration option, eg -o dir::cache=/tmp
See the apt-get(8), sources.list(5) and apt.conf(5) manual
pages for more information and options.

This APT has Super Cow Powers.

```

Le `-h` est parfois un bon complément au `man` si vous n'arrivez pas à comprendre comment utiliser la commande. On y trouve parfois des informations utiles comme ici : « *The most frequently used commands are update and install* », ce qui signifie que l'on utilise le plus souvent `apt-get` avec les paramètres `update` et `install` (et c'est vrai).



Parfois, il n'y a pas de page de manuel pour une commande ( `man` ne fonctionne pas pour cette dernière) mais le `-h` ou le `--help` fonctionne. Pensez-y !

## La commande `whatis`

La commande `whatis` est une sorte de `man` très allégé. Elle donne juste l'en-tête du manuel pour expliquer en deux mots à quoi sert la commande. Par exemple pour `mkdir` :

```
whatis mkdir
```

Ça vous permet d'éviter de sortir l'artillerie lourde juste pour savoir à quoi sert la commande.

```
mateo21@mateo21-desktop:~$ whatis mkdir
mkdir (1)             - make directories
```

## Rechercher `man` sur le Web

Enfin, il est bien de le préciser : on retrouve aussi le `man` sur le Web !

Si vous devez lire un manuel et que vous n'êtes pas sous Linux à ce moment-là, utilisez tout bêtement un moteur de recherche comme Google pour retrouver la doc.

Par exemple, vous pouvez taper la recherche : `man mkdir` .

Je vous parie que vous trouverez le manuel dans les premiers liens qui s'offrent à vous.

Bonne recherche !

## En résumé

- Sous Linux, toutes les commandes et leurs paramètres sont documentés dans le manuel. Il est recommandé de lire le manuel à chaque fois que vous avez des questions sur une commande car la réponse s'y trouve la plupart du temps.
- On fait appel au manuel avec la commande `man` suivie du nom de la commande sur laquelle on veut avoir plus d'informations. Par exemple : `man mkdir` .
- Dans le manuel, on se déplace avec les touches fléchées ou `Page Up` et `Page Down` , on fait une recherche avec la touche `/` (slash) et on quitte avec la touche `Q` .

## Le professeur



**Mathieu Nebra**

Le manuel d'une commande commence toujours par son `SYNOPSIS` : c'est un résumé des différentes manières d'utiliser la commande. Les options facultatives sont écrites entre crochets.

ur trouver une commande correspondant à un certain usage, utilisez `apropos` . Ainsi, `apropos sound` affichera toutes les commandes ayant un rapport avec le son.

Entrepreneur à plein temps, auteur à plein temps et co-fondateur d'OpenClassrooms :o)

J'AI TERMINÉ CE CHAPITRE ET JE PASSE AU SUIVANT

## Découvrez aussi ce cours en...



Livre



PDF

INSTALLER DES PROGRAMMES AVEC APT-GET

RECHERCHER DES FICHIERS



## 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 

