

GNU - Linux // Commande : tldr (Too Long; Didn't Read)

Présentation de l'Outil.....	2
"Pourquoi lire le manuel quand on peut avoir la réponse ?".....	2
Processus d'Installation.....	2
Utilisation Concrète (Ou "Comment tricher légalement").....	4
Le Principe.....	4
Cas Pratique N°1 : L'Enfer des Archives (tar).....	4
Cas Pratique N°2 : La Recherche d'Aiguille (find).....	5
Cas Pratique N°3 : Le Réseau (ssh).....	6
Cas Pratique N°4 : Le Réseau avec ip.....	7
Le Cas du "Multivers" (Option -platform).....	8
Maintenance de l'Outil.....	9

Présentation de l'Outil

"Pourquoi lire le manuel quand on peut avoir la réponse ?"

Note : Comme dirait Kevin dans *The Office* : "Why waste time say lot word when few word do trick?". Cette documentation est faite pour vous si votre capacité de concentration est inférieure à celle d'un poisson rouge devant un économiseur d'écran.

TLDR est une collection de pages d'aide simplifiées et communautaires. Contrairement à man (qui est exhaustif, technique et aride), tldr se concentre sur **l'exemple concret**.

- **Man :** Vous explique la théorie, la syntaxe, l'histoire et les 400 options.
- **TLDR :** Vous donne la commande exacte que vous allez copier-coller.

Processus d'Installation

Nous allons installer la version la plus stable. Selon votre distribution, la méthode varie légèrement, mais restons sur du classique.

La méthode Python :

On s'assure d'avoir les dépendances

```
admin@lab-debian-21:~$ sudo apt install python3-pip
[sudo] password for admin:
Installing:
  python3-pip

Installing dependencies:
  binutils                g++-14                libasan8                libfile-fcntllock-perl  libmpc3
  make                    binutils-common       g++-14-x86-64-linux-gnu libbinutils              libgcc-14-dev           libmpfr6
  manpages-dev            binutils-x86-64-linux-gnu g++-x86-64-linux-gnu  libc-dev-bin            libgomp1                libpython3-dev
  patch                   build-essential       gcc                     libc6-dev                libgprofng0             libpython3.13
  python3-dev              cpp                    gcc-14                  libcc1-0                 libhwasan0              libpython3.13-
  dev python3-packaging    cpp-14                 gcc-14-x86-64-linux-gnu libcrypt-dev              libisl23                libquadmath0
  python3-wheel            cpp-14-x86-64-linux-gnu gcc-x86-64-linux-gnu  libctf-nobfd0            libitm1                 libsframe1
  python3.13-dev           cpp-x86-64-linux-gnu  javascript-common     libctf0                  libjs-jquery            libstdc++-14-dev
  rpcsvc-proto            dpkg-dev              libalgorithm-diff-perl libdpkg-perl             libjs-sphinxdoc        libtsan2
  sq                       fakeroot              libalgorithm-diff-xs-perl libexpat1-dev            libjs-underscore       libubsan1
  zlib1g-dev               g++                   libalgorithm-merge-perl libfakeroot              liblsan0                linux-libc-dev

Suggested packages:
  binutils-doc  cpp-14-doc  gcc-14-doc  flex  gdb-x86-64-linux-gnu  glibc-doc
  ed            gprofng-gui  debian-keyring  gcc-multilib  bison  apache2  git
  diffutils-doc binutils-gold  debian-tag2upload-keyring  autoconf  gdb  |  httpd  bzip2
  python3-setuptools  cpp-doc  g++-multilib  automake  gcc-doc  |  httpd  libstdc++-14-
  doc              gcc-14-locales  g++-14-multilib  libtool  gcc-14-multilib  libc-devtools  make-doc

Summary:
  Upgrading: 0, Installing: 66, Removing: 0, Not Upgrading: 5
  Download size: 93.0 MB
  Space needed: 361 MB / 22.3 GB available

Continue? [Y/n]
```

On installe le gestionnaire d'environnements isolés

```
admin@lab-debian-21:~$ sudo apt install pipx
Installing:
  pipx

Installing dependencies:
  python3-argcomplete  python3-pip-whl      python3-psutil      python3-userpath  python3.13-venv
  python3-click         python3-platformdirs  python3-setuptools-whl  python3-venv

Summary:
  Upgrading: 0, Installing: 10, Removing: 0, Not Upgrading: 5
  Download size: 3,994 kB
  Space needed: 8,602 kB / 21.9 GB available

Continue? [Y/n]
```

On installe tldr via pipx

```
admin@lab-debian-21:~$ sudo pipx install tldr
installed package tldr 3.4.3, installed using Python 3.13.5
These apps are now globally available
- tldr
These manual pages are now globally available
- man1/tldr.1
⚠ Note: '/root/.local/bin' is not on your PATH environment variable. These apps will not be globally
accessible until your PATH is updated. Run
`pipx ensurepath` to automatically add it, or manually modify your PATH in your shell's config file (e.g.
~/.bashrc).
done! ✨ ✨ ✨
```

(Important) On s'assure que le chemin est bon

```
admin@lab-debian-21:~$ sudo pipx ensurepath
Success! Added /root/.local/bin to the PATH environment variable.

Consider adding shell completions for pipx. Run 'pipx completions' for instructions.

You will need to open a new terminal or re-login for the PATH changes to take effect. Alternatively, you can
source your shell's config file with
e.g. 'source ~/.bashrc'.

Otherwise pipx is ready to go! ✨ ✨ ✨
```

Mise à jour du cache (OBLIGATOIRE)

```
admin@lab-debian-21:~$ source ~/.bashrc
```

```
admin@lab-debian-21:~$ tldr --update
Updated cache for language en: 6818 entries
Updated cache for language en: 6818 entries
```

Utilisation Concrète (Ou "Comment tricher légalement")

Note : C'est ici que la magie opère. Là où man vous donne les plans de l'architecte, tldr vous donne la clé de la maison. C'est le moment "I know Kung Fu" de Matrix.

Le Principe

La syntaxe est enfantine : `tldr <nom_de_la_commande> (facultatif <option_de_la_commande>)`

Cas Pratique N°1 : L'Enfer des Archives (tar)

Personne, je dis bien *personne*, ne se souvient des options de tar par cœur.

- **Version man** : Une lecture de 20 minutes sur les tailles de blocs et les normes POSIX.
- **Version tldr tar** :

```
admin@lab-debian-21:~$ tldr tar
```

tar

Archiving utility.
Often combined with a compression method, such as `gzip` or `bzip2`.
More information: <https://www.gnu.org/software/tar/manual/tar.html>.

- [c]reate an archive and write it to a [f]ile:
`tar cf path/to/target.tar path/to/file1 path/to/file2 ...`
- [c]reate a g[z]ipped archive and write it to a [f]ile:
`tar czf path/to/target.tar.gz path/to/file1 path/to/file2 ...`
- [c]reate a g[z]ipped (compressed) archive from a directory using relative paths:
`tar czf path/to/target.tar.gz --directory path/to/directory .`
- E[x]tract a (compressed) archive [f]ile into the current directory [v]erbosely:
`tar xvf path/to/source.tar[.gz|.bz2|.xz]`
- E[x]tract a (compressed) archive [f]ile into the target directory:
`tar xf path/to/source.tar[.gz|.bz2|.xz] --directory path/to/directory`
- [c]reate a compressed archive and write it to a [f]ile, using the file extension to [a]utomatically determine the compression program:
`tar caf path/to/target.tar.xz path/to/file1 path/to/file2 ...`
- Lis[t] the contents of a tar [f]ile [v]erbosely:
`tar tvf path/to/source.tar`
- E[x]tract files matching a pattern from an archive [f]ile:
`tar xf path/to/source.tar --wildcards "*.html"`

Cas Pratique N°2 : La Recherche d'Aiguille (find)

La commande find est puissante mais sa syntaxe est aussi accueillante qu'un videur à l'entrée d'une boîte de nuit.

- **Commande** : tldr find

```
admin@lab-debian-21:~$ tldr find
```

find

Find files or directories under a directory tree, recursively.
See also: `fd`.
More information: <https://manned.org/find>.

- Find files by extension:
`find path/to/directory -name '*.ext'`
- Find files matching multiple path/name patterns:
`find path/to/directory -path '*/path/*/*.ext' -or -name '*pattern*'`
- Find directories matching a given name, in case-insensitive mode:
`find path/to/directory -type d -iname '*lib*'`
- Find files matching a given pattern, excluding specific paths:
`find path/to/directory -name '*.py' -not -path '*/site-packages/*'`
- Find files matching a given size range, limiting the recursive depth to "1":
`find path/to/directory -maxdepth 1 -size +500k -size -10M`
- Run a command for each file (use {} within the command to access the filename):
`find path/to/directory -name '*.ext' -exec wc -l {} \;`
- Find all files modified today and pass the results to a single command as arguments:
`find path/to/directory -daystart -mtime -1 -exec tar -cvf archive.tar {} \+`
- Search for either empty files or directories and delete them verbosely:
`find path/to/directory -type f|d -empty -delete -print`

Found 1 page with the same name under the platform: windows.

Cas Pratique N°3 : Le Réseau (ssh)

Pour se connecter sans réfléchir aux clés et aux ports.

- **Commande** : tldr ssh

```
admin@lab-debian-21:~$ tldr ssh

ssh

Secure Shell is a protocol used to securely log onto remote systems.
It can be used for logging or executing commands on a remote server.
More information: https://man.openbsd.org/ssh.

- Connect to a remote server:
  ssh username@remote_host

- Connect to a remote server with a specific [i]dentity (private key):
  ssh -i path/to/key_file username@remote_host

- Connect to a remote server with IP 10.0.0.1 and using a specific [p]ort (Note: 10.0.0.1 can be shortened to 10.1):
  ssh username@10.0.0.1 -p 2222

- Run a command on a remote server with a [t]ty allocation allowing interaction with the remote command:
  ssh username@remote_host -t command command_arguments

- SSH tunneling: [D]ynamic port forwarding (SOCKS proxy on localhost:1080):
  ssh -D 1080 username@remote_host

- SSH tunneling: Forward a specific port (localhost:9999 to example.org:80) along with disabling pseudo-[T]ty allocation and executio[N] of remote commands:
  ssh -L 9999:example.org:80 -N -T username@remote_host

- SSH [J]umping: Connect through a jumphost to a remote server (Multiple jump hops may be specified separated by comma characters):
  ssh -J username@jump_host username@remote_host

- Close a hanged session:
  <Enter><-><.>
```

Cas Pratique N°4 : Le Réseau avec ip

- Commande : tldr ip

```
admin@lab-debian-21:~$ tldr ip

ip

Show/manipulate routing, devices, policy routing and tunnels.
Some subcommands such as `address` have their own usage documentation.
More information: https://manned.org/ip.8.

- List interfaces with detailed info:
  ip address

- List interfaces with brief network layer info:
  ip -brief address

- List interfaces with brief link layer info:
  ip -brief link

- Display the routing table:
  ip route

- Show neighbors (ARP table):
  ip neighbour

- Make an interface up/down:
  sudo ip link set ethX up|down

- Add/Delete an IP address to an interface:
  sudo ip address add|delete ip_address/mask dev ethX

- Add a default route:
  sudo ip route add default via ip_address dev ethX
```

- Commande : tldr ip route

```
admin@lab-debian-21:~$ tldr ip route

ip route

IP Routing table management subcommand.
See also: `routel`.
More information: https://manned.org/ip-route.

- Display the main routing table:
  ip route

- Add a default route using gateway forwarding:
  sudo ip route add default via gateway_ip

- Add a default route using ethX:
  sudo ip route add default dev ethX

- Add a static route:
  sudo ip route add destination_ip via gateway_ip dev ethX

- Delete a static route:
  sudo ip route delete destination_ip dev ethX

- Change or replace a static route:
  sudo ip route change|replace destination_ip via gateway_ip dev ethX

- Show which route will be used by the kernel to reach an IP address:
  ip route get destination_ip

- Display a specific routing table:
  ip route list table table_number
```

Le Cas du "Multivers" (Option `-platform`)

Parfois, une commande fonctionne différemment sous Linux, Mac (OSX) ou Windows. `tar` sous Mac n'est pas le même que sous Linux (BSD vs GNU).

Si vous êtes sur Linux mais que vous voulez voir la commande pour un collègue sur Mac (pour vous moquer de lui, bien sûr) :

Affiche la version OSX de la commande

```
admin@lab-debian-21:~$ tldr -p osx tar
```

`tar`

Archiving utility.
Often combined with a compression method, such as ``gzip`` or ``bzip2``.
More information: <https://www.gnu.org/software/tar/manual/tar.html>.

- [c]reate an archive and write it to a [f]ile:
`tar cf path/to/target.tar path/to/file1 path/to/file2 ...`
- [c]reate a g[z]ipped archive and write it to a [f]ile:
`tar czf path/to/target.tar.gz path/to/file1 path/to/file2 ...`
- [c]reate a g[z]ipped (compressed) archive from a directory using relative paths:
`tar czf path/to/target.tar.gz --directory path/to/directory .`
- E[x]tract a (compressed) archive [f]ile into the current directory [v]erbosely:
`tar xvf path/to/source.tar[.gz|.bz2|.xz]`
- E[x]tract a (compressed) archive [f]ile into the target directory:
`tar xf path/to/source.tar[.gz|.bz2|.xz] --directory path/to/directory`
- [c]reate a compressed archive and write it to a [f]ile, using the file extension to [a]utomatically determine the compression program:
`tar caf path/to/target.tar.xz path/to/file1 path/to/file2 ...`
- Lis[t] the contents of a tar [f]ile [v]erbosely:
`tar tvf path/to/source.tar`
- E[x]tract files matching a pattern from an archive [f]ile:
`tar xf path/to/source.tar --wildcards "*.html"`

Maintenance de l'Outil

tldr est communautaire. De nouvelles fiches sont ajoutées tous les jours par des gens qui, comme vous, en ont eu marre de chercher.

Pensez à rafraîchir le cache une fois par mois, sinon vous risquez de rater des mises à jour critiques.

```
admin@lab-debian-21:~$ tldr --update
Updated cache for language en: 6818 entries
Updated cache for language en: 6818 entries
```