

Linux - commandes fondamentales

Un petit guide pour bien démarrer avec Linux.

ls cd cp mv rm

mkdir chmod chown

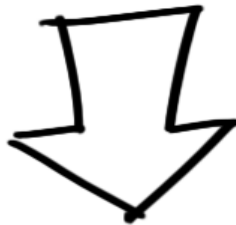
ssh scp rsync sed grep find test

Paquets Processus Archives Espace disque

[Tweeter](#)

Par Julien Dubreuil

Inspiré du guide de Roger Dudler sur Git.



ls

La commande **ls** permet d'afficher le contenu d'un répertoire.

ls -l Afficher les informations de manière détaillée.

ls -a Afficher les fichiers cachés.

`ls -h` Afficher la taille des fichiers de façon lisible.

`ls -r` Tri inversé.

`ls -t` Trier les fichiers par date du plus récent au plus ancien.

`ls -S` Trier par taille décroissante.

`ls -la` Afficher tous les fichiers y compris les fichiers cachés.

`ls -lhS` Afficher les informations des fichiers, avec des tailles lisibles le tout ordonné du plus grand au plus petit.

cd

La navigation d'un répertoire à un autre s'effectue avec la commande ***cd*** suivie du nom du répertoire.

`cd /` Permet de se retrouver à la racine du disque.

`cd ~` ou `cd` Accéder directement au répertoire de l'utilisateur.

`cd /var/www/` Aller dans le répertoire `/var/www`.

`cd ..` Remonter dans le répertoire parent à partir de là où vous êtes.

`cd -` Permet de revenir au répertoire précédent.

`pwd` Renvoyer le chemin absolu du répertoire courant ce qui est utile puisqu'en général le shell n'affiche que le nom du répertoire courant.

cp

Pour créer une copie d'un fichier, on utilise la commande **cp**.

`cp foo/bar.txt baz/` Copier le fichier bar.txt dans le répertoire baz.

`cp -r foo/ baz/` Copier des répertoires entiers (note : si baz existe, la cible sera baz/foo/).

mv

Déplacer un fichier est aussi simple que de le copier, pour cela il faut utiliser la commande **mv**. Cette commande permet aussi de renommer vos fichiers.

`mv foo/bar.txt baz/` Déplacer le fichier bar.txt dans le répertoire baz.

`mv foo_bar.txt foo_baz.txt` Renommer le fichier foo_bar.txt en foo_baz.txt.

rm

Pour effacer un fichier ou un répertoire on utilise la commande ***rm***.

`rm *.txt` Supprimer tous les fichiers ayant pour extension txt.

`rm foo.txt bar.txt` Supprimer les fichiers foo.txt et bar.txt.

`rm -rf baz/` Supprimer le répertoire baz et tout son contenu.

mkdir

Pour créer un répertoire il suffit d'utiliser la commande ***mkdir***. Celle-ci vous permettra de créer un répertoire à l'emplacement où vous êtes ou l'emplacement précisé en argument de la commande.

`mkdir -v` Retourner des informations lors de la création d'un répertoire.

`mkdir -p` Cette option permet de créer une arborescence complète.

`mkdir foo` Créer le répertoire foo.

`mkdir -v foo /tmp/bar` Créer les répertoires foo et /tmp/bar.

`mkdir -p foo/bar/baz` Créer l'arborescence foo/bar/baz.

chown

La commande **chown** permet de changer le propriétaire d'un fichier ou d'un répertoire.

`chown bob:admin foo.txt` Attribuer l'utilisateur bob et le groupe admin au fichier foo.txt.

chmod

La commande **chmod** permet de changer les droits d'un fichier ou d'un répertoire. Pour exécuter cette commande vous devez être le propriétaire du fichier ou être logué en root.

`chmod u+w fichier` Ajouter les droits d'écriture au propriétaire (user, write)

`chmod g+r fichier` Ajouter les droits de lecture au groupe du fichier (group, read)

`chmod o-x fichier` Supprimer les droits d'exécution aux autres utilisateurs (other, execution)

`chmod a+rw dossier` Ajouter les droits de lecture / écriture à tous (all)

`chmod -R a+rx files` Ajouter les droits de lecture et d'exécution à tout ce que contient le repertoire dossier.

`chmod 764 dossier` Tous les droits pour le propriétaire (7xx), lecture et écriture pour le groupe (x6x) et lecture uniquement pour les autres (xx4).

`chmod -R 755 dossier` Donner au propriétaire tous les droits (7xx), alors que seuls les droits de lecture et d'accès seront donnés aux autres (x55). Grace à l'option -R ces droits seront appliqués à tous les fichiers et dossiers contenus dans ce répertoire.

Correspondances de représentation des droits

Droit	Valeur alphanumérique	Valeur octale
aucun droit	---	0
exécution seulement	--X	1
écriture seulement	-W-	2
écriture et exécution	-WX	3
lecture seulement	r--	4
lecture et exécution	r-X	5
lecture et écriture	rw-	6
tous les droits (lecture, écriture et exécution)	rwX	7

ssh

La commande **ssh** permet de se connecter de façon sécurisée à une machine distante.

`ssh john@remotehost.example.com` Connexion à la machine distante avec le login john.

`ssh -l john remotehost.example.com` Equivaut à la commande précédente.

`ssh-keygen -t dsa` Génération d'une clé DSA (à faire sur la machine locale).

`ssh-copy-id -i ~/.ssh/id_dsa.pub john@remotehost.example.com` Copie de la clé publique sur la machine distante.

scp

La commande **scp** permet de copier des fichiers entre le serveur et le client ssh de manière sécurisée.

`scp foo.txt john@remotehost.example.com:` Transfère le fichier foo.txt situé dans le répertoire courant vers le home du compte john de la machine remotehost.example.com.

`scp john@remotehost.example.com:foo.txt ./` Récupère le fichier foo.txt situé dans le home du répertoire du compte john pour le copier dans le répertoire courant.

`scp john@remotehost.example.com:/backups/*.sql backups/` Récupérer les fichiers .sql situés dans le répertoire backups pour le copier dans le sous-répertoire backups.

`scp -P 17654 john@remotehost:/files/ files/` Récupérer les fichiers via un autre port (17654) que le port par défaut (22).

`scp -r mails/ john@remotehost:` Transfère l'intégralité du répertoire mails.

rsync

Grâce à la commande ***rsync*** vous pouvez copier des fichiers et des dossiers depuis ou vers un hôte à distance. Grâce à cela vous pouvez faire de la synchronisation de fichiers.

Quelques options :

- a: copie de manière récursive tout en préservant les permissions et les dates des fichiers.
- z: compresse les données avant le transfert.
- v: affiche tout ce qui se passe pendant le transfert.
- n: permet de tester la commande (*dry-run*).
- progress: affiche la barre de progression pendant le transfert.
- delete: efface les fichiers qui n'existent pas chez l'émetteur
- exclude=MOTIF: exclut les fichiers correspondant au motif

`rsync source/ destination/` Synchroniser les fichiers sources vers une destination.

`rsync -azv john@remotehost.example.com:source/ destination/` Récupère les nouveaux fichiers et fichiers modifiés du répertoire distant vers le répertoire local.

`rsync -azv --delete source/ destination/` Synchroniser tous les fichiers, supprime aussi les fichiers qui n'existent plus sur la source.

`rsync -azv --exclude="dump/*.sql.gz" files/ backup/` Synchroniser tout le répertoire files sauf les fichiers .sql.gz.

`rsync --rsh='ssh -p1234' source/ destination/` Effectuer un rsync sur un port spécifique.

Note : en général, on suffixe toujours par un `/` le nom des répertoires sources et de destination.

sed

La commande **sed** est un utilitaire qui parcourt un fichier texte ligne par ligne afin de lui appliquer un traitement ou un remplacement lorsque l'expression régulière est vérifiée.

`sed 's/foo/bar/' file.txt` Transformer la première occurrence 'foo' de chaque ligne par 'bar'.

`sed 's/\t/ /g' file.txt` Transformer toutes les tabulations par deux espaces.

`sed '/^#/ d' file.txt` Supprimer toutes les lignes commençant par #.

`sed '/^Bonjour/,/^Au revoir/d' file.txt` Supprimer toutes les lignes comprises entre les deux motifs.

`sed -n '/foo/p' file.txt` Afficher uniquement les lignes où l'expression est trouvée.

grep

La commande **grep** permet de rechercher une chaîne de caractères ou un motif dans un fichier.

Quelques options :

- v: affiche les lignes ne contenant pas la chaîne.
- c: compte le nombre de lignes contenant la chaîne.
- n: retourne les lignes préfixées par leur numéro.
- x: ligne correspondant exactement à la chaîne.
- l: affiche le nom des fichiers qui contiennent la chaîne.

`grep 'text' foo.txt` Recherche l'occurrence 'text' dans le fichier foo.txt.

`grep -nri 'foobar' /project` Recherche toutes les occurrences de 'foobar' dans le repertoire /project.

`grep -nri '\(foo\|bar\|baz\)' /project` Recherche toutes les occurrences à 'foo', 'bar' et 'baz' dans le repertoire /project.

find

La commande ***find*** permet de chercher des fichiers et éventuellement d'exécuter des commandes dessus.

Quelques options :

- name*: Recherche d'un fichier par son nom
- iname*: Même chose que -name mais insensible à la casse
- type*: Recherche de fichier d'un certain type
- atime*: Recherche par date de dernier accès
- mtime*: Recherche par date de dernière modification
- user*: Recherche de fichiers appartenant à l'utilisateur donné
- group*: Recherche de fichiers appartenant au groupe donné
- size*: Recherche par rapport à une taille de fichier.
- exec*: Exécute la commande donnée aux fichiers trouvés.
- a*: Opérateur ET
- o*: Opérateur OU
- ! ou -*not*: Opérateur NOT

`find myfile* -print` Rechercher un fichier commençant par "myfile"

`find -name *myfile*.txt -print` Rechercher un fichier contenant "myfile" et ayant pour extension ".txt"

`find /usr -type d -print` Afficher tous les répertoires de /usr

`find $HOME \(-name '*.txt' -o -name '*.pdf' \)` Afficher tous les fichiers .txt ou .pdf dans le répertoire home de l'utilisateur.

`find $HOME -name *.txt -atime +7 -exec rm {} \;` Supprimer tous les fichiers .txt qui n'ont pas été consultés depuis plus de 7 jours dans le répertoire home de l'utilisateur.

```
find $HOME -name '*.txt' -size +4k -exec ls -l {} \;
```

[Afficher la taille de tous les fichiers de plus de 4 kilos](#)

test

La commande **test** permet comparer des chaînes de caractères, des nombres et vérifier certaines propriétés de fichiers. Il est possible de simplifier son écriture en la remplaçant par des crochets [].

```
test -e foo.txt
```

 tester si le fichier foo.txt existe.

```
test -d foo
```

 tester que foo soit un répertoire.

```
test -w foo.txt
```

 tester si le fichier est accessible en écriture.

```
test -x foo.txt
```

 tester si le fichier est exécutable.

```
[ "string1" != "string2" ]
```

 vérifier que la chaîne string1 n'est pas égale à string2.

```
[ int1 != int2 ]
```

 vérifier que le nombre int1 est inférieur à int2.

packages

L'installation et la désinstallation de paquets sous Debian, Ubuntu et distribution dérivées peut se faire grâce à la commande ***apt***.

`apt-get update` Mettre à jour la liste des fichiers disponibles dans les dépôts APT.

`apt-get install samba` Installer du paquet Samba.

`apt-get install foo=2.2-1` Installer du paquet foo dans sa version 2.2-1.

`apt-get remove samba` Désinstallation du paquet Samba tout en laissant les fichiers de configuration.

`apt-get purge samba` Suppression complète du paquet Samba et de ses fichiers de configuration.

`apt-cache policy php5` Récupération d'informations sur l'état du paquet php5

`dpkg -l | grep php` Lister tous les paquets php installés sur la machine

archives

Que ce soit pour compresser ou décompresser des fichiers ou des dossiers, cela se fait avec la commande ***tar***.

Quelques options :

- c: créer
- t: tester / lister
- x: extraire
- v: description des fichiers désarchivés
- j: format de compression bzip2
- z: format de compression gzip

`tar -cvf archive.tar fichier1` Création d'une archive nommée `archive.tar` contenant le fichier `fichier1`.

`tar -cvf archive.tar fichier1 fichier2` Création d'une archive contenant deux fichiers `fichier1` et `fichier2`.

`tar -cvf archive.tar repertoire/` Création d'une archive a partir d'un répertoire.

`tar -czvf archive.tar.gz repertoire/` Création d'une archive au format `tar.gz`.

`tar -cjvf archive.tar.bz2 repertoire/` Création d'une archive au format `tar.bz2`.

`tar -xzvf archive.tar.gz` Extraction de l'archive `tar.gz`.

`tar -xjvf archive.tar.bz2` Extraction de l'archive `tar.bz2`.

`tar -tf mon_fichier.tar` Liste tous les fichiers contenus dans une archive.

espace disque

`du -sh dossier1 dossier2` connaitre l'espace disque utilisé des deux répertoires (*disk usage*).

`du -hc --max-depth=1` afficher l'espace disque utilisé des fichiers et répertoires contenu dans un répertoire.

`df -h` afficher l'espace disque disponible (*disk free*).

gestion des processus

`top` Classement en live des processus en cours triés par utilisation Proc, Mem ou Temps CPU.

`free` Afficher la mémoire libre.

`ps aux` Afficher tous les processus exécutés.

`ps faux` Afficher tous les processus exécutés affiché sous forme.

`kill pid` Arrêter un processus.

`kill 9 pid` Tuer violemment le processus (déconseillé).

commentaires

50 Comments

Linux - commandes fondamentales

 Login ▾ Recommend 16 Share

Sort by Best ▾



Join the discussion...

LOG IN WITH

OR SIGN UP WITH DISQUS 

Eric • 5 years ago

Bonjour, présentation et explications claires, bravo pour ce blog. Pourriez-vous ajouter des commandes système ? par exemple : de combien de mémoire mon poste dispose, quelles sont les capacités disque, ..etc..
Bonne continuation.

4 ^ | ▾ • Reply • Share ›

**Eddy** • 3 years ago

Comment voir les droits du répertoire dans lequel on se trouve sans se déplacer?

2 ^ | v • Reply • Share ›

Kyrax ➔ Eddy • 3 years ago

C'est bourrin mais ça fonctionne...

```
root@server:/home# ll .. | grep ${PWD##*/}
drwxr-xr-x 6 root root 4096 déc. 18 11:11 home
```

^ | v • Reply • Share ›

Julien Verkest • 4 years ago

Très pratique et très joli, merci !

2 ^ | v • Reply • Share ›

howayda 299 • 3 months ago

where commande "cat" plz give des exemple

1 ^ | v • Reply • Share ›

**Joel Amouzou** • 3 years ago

Readable and Well explained ! Congratulation.... But you must improve with more commands

1 ^ | v • Reply • Share ›

Julien Dubreuil Mod ➔ Joel Amouzou • 3 years ago

Hi Joel. Thanks for your feedback. If you have suggestions feel free to suggest them :)

^ | v • Reply • Share ›

Nehemie KOFFI • 4 years ago

Super! ça permet de gagner en temps. Good inspiration!

1 ^ | v • Reply • Share ›

†‡!0‡!‡! XX • 23 days ago

Très claire! merci

^ | v • Reply • Share ›

Oussama MLeiki • 3 months ago

tnx

^ | v • Reply • Share ›

**A.H** • 4 months ago

T'aurais pu rajouter la fonction touch qui permet de créer un fichier

^ | v • Reply • Share ›

**Lilian** • 5 months ago

Bien présenté et surtout bien pratique, merci !

^ | v • Reply • Share ›

Jean-Baptiste • 9 months ago

Bonjour, merci pour cette belle présentation. Au fait, y-at-il une commande pour avoir la liste complète des commandes de bases?

^ | v • Reply • Share ›

vincent • 9 months ago

Un très bon site, mais où trouver les tampons de la console ?

^ | v • Reply • Share ›

Euloge konan • a year ago

Bonjour, si vous plaît je cherche une commande pour restaurer ma machine debian, je veux tout mettre à zero, en un mots je veux grâce à la commande avoir une machine neuve. merci bien

^ | v • Reply • Share ›

**baboum** • a year ago

Salut Julien,

Superbe article. Des suggestions, si tu es motivé tu peux aussi ajouter "mysqldump", elle est plutôt facile à utiliser et très pratique. Et

aussi l'utilisation de "crontab".

^ | v • Reply • Share ›



majucarma • 2 years ago

Très bon début. Il ne faut pas s'arrêter là. Cela vaut la peine de compléter avec des commande un peu plus élémentaires pour "utiliser" Linux telle que lsmod, lspci.

En tout cas bravo, ce site est clair et précis.

^ | v • Reply • Share ›

Geoffroy • 2 years ago

Cette page va dans mes favoris tout de suite!

Merci beaucoup :)

^ | v • Reply • Share ›

Alex Commans • 2 years ago

bonjour Très chouette Tuto j'ai appris beaucoup de choses mais Je ne suis pas le plus doué ... alors j'ai une petite question, je travaille sur Mac je voudrais transférer un dossier que j'ai sur mon server chez OVH vers mon disque dur OSX ... si quelqu'un peut m'aider d'avance merci Alex

^ | v • Reply • Share ›

mimi • 2 years ago

comment sourcer l'envennement?

la commande qui affiche l'état de package

^ | v • Reply • Share ›



Hakizimana Jean Marie • 2 years ago

Merci pour nous enseigner!

^ | v • Reply • Share ›

claudi • 2 years ago

cool, j'ai aimé ce tuto

^ | v • Reply • Share ›

O.Geoffrey • 2 years ago

Super résumé, très utile et bien mis en page ! Merci, toujours utile, même si le man existe.

^ | v • Reply • Share ›

Slim Ltaief • 2 years ago

bravo ! c'est un bon travail..merci bien :)

^ | v • Reply • Share ›



Thomas • 2 years ago

Bonjour,

Félicitations pour ce mémo très clair.

Cependant, il y a une erreur pour la commande "chmod", c'est pour changer les droits sur un fichier ou un répertoire et non pour changer de propriétaire comme le fait la commande "chown" juste au dessus.

^ | v • Reply • Share ›

Julien Dubreuil Mod ➔ **Thomas** • 2 years ago

Oh oui effectivement, merci c'est maintenant corrigé :) Ce petit guide est un document communautaire, n'hésitez pas à soumettre des pull request pour ajouter des informations !

^ | v • Reply • Share ›



Sebelectronique • 2 years ago

Merci j'ajoute votre blog sur mon blog ! lol @+ <http://sebelectronique.cana...>

^ | v • Reply • Share ›



David • 2 years ago

Super tutoriel, notamment la présentation, certes reprise, très agréable.

kill 9 <pid> ne marche toutefois pas sous bash, il faut écrire -9

^ | v • Reply • Share ›

Loubna Chahid • 2 years ago

merci c est plus détailler

^ | v • Reply • Share ›



Jeanjean • 3 years ago

Merci pour l'article Julien. Il y a juste une petite erreur au paragraphe "chmod".
Tu écris la commande "chown" au lieu de "chmod".

^ | v • Reply • Share ›

Julien Dubreuil Mod ➔ Jeanjean • 3 years ago

Oh bien vu ! Merci Jeanjean

^ | v • Reply • Share ›

Stev Crow • 3 years ago

Merci (y) :)

^ | v • Reply • Share ›



yane • 4 years ago

très joli blog ;très intéressant.bravo et merci.

^ | v • Reply • Share ›



Charly • 4 years ago

Merci!!!

^ | v • Reply • Share ›



Mohcine • 4 years ago

Perfect !!

^ | v • Reply • Share ›

yassine • 4 years ago

merci bk pour ces information que dieux vous garde

^ | v • Reply • Share ›



Pierre • 4 years ago

J'arrive de ton autre (ou plutôt: d'un de tes autres...) sites.. et celui ci aussi il est cool. Très original. (-:

^ | v • Reply • Share ›



Arl@nd • 4 years ago

i like this page!!!!

^ | v • Reply • Share ›



khadija • 4 years ago

merci beaucoup pour ces commandes

^ | v • Reply • Share ›



yassine • 4 years ago

thanks

^ | v • Reply • Share ›



badich • 4 years ago

Merci beaucoup

^ | v • Reply • Share ›



Beninho • 4 years ago

c'est cool merci de me rafraîchir la mémoire

^ | v • Reply • Share ›



tizafri • 4 years ago

Merci beaucoup pour ces commandes

^ | v • Reply • Share ›

ptitboby • 5 years ago

Super boulot et assez pratique, les commandes principal y sont .

ne pas oublier la commande man pour avoir plus d'option sur une commande.

Merci encore

^ | v • Reply • Share ›



charbelsar • 5 years ago

comment ouvrir un fichier avec le terminal

^ | v • Reply • Share ›



carl → charbelsar • 4 years ago

t'as pour cela plusieurs outils :

```
# vi /chemin/vers/mon/fichier
# vim /chemin/vers/mon/fichier
# nano /chemin/vers/mon/fichier
```

etc.

^ | v • Reply • Share ›

Blandin • 5 years ago

Bah scp, ça m'a bien servi, surtout pour transférer des dumps de bases de données entre les serveurs de prod et les serveurs de recette. L'autre truc qui me sert bien, c'est de lancer une commande et de la détacher de ma console, ça évite qu'une coupure de connexion internet ne coupe la restauration d'une base de données de 40Go (oui, j'étais vert quand c'est arrivé).

Donc, on lance la commande, après, un petit ctrl-z (qui met en pause la commande), puis on lance la commande bg (qui reprend la commande et la fait tourner en arrière plan) puis un disown (qui détache la commande de la console qui l'a lancée), du coup, la commande continue même si l'on perd la connexion ou que l'on ferme la console.

Bon, c'est surement moins utilisé, mais quand on a une commande longue à lancer et que l'on veut continuer à faire quelque chose d'autre sans ouvrir une autre console, c'est pratique.

^ | v • Reply • Share ›



carl → Blandin • 4 years ago

plus facile encore pour lancer une commande en background c'est la terminer par un "&" :

```
# commande &
```

^ | v • Reply • Share ›



David → carl • 2 years ago

Sauf que cela ne la détache pas de la console, elle travaille en arrière-plan dans la console courante.

Le plus simple pour détacher une commande est de la préfixer de "nohup": "nohup firefox" par exemple, ce qui permet de fermer la console sans que firefox se ferme aussi

^ | v • Reply • Share ›



Baptou • 5 years ago

Super petit memento quand on utilise pas linux tous les jours..

^ | v • Reply • Share ›