

Antisèche commandes Raspberry Pi

74

Commandes pour Raspberry Pi

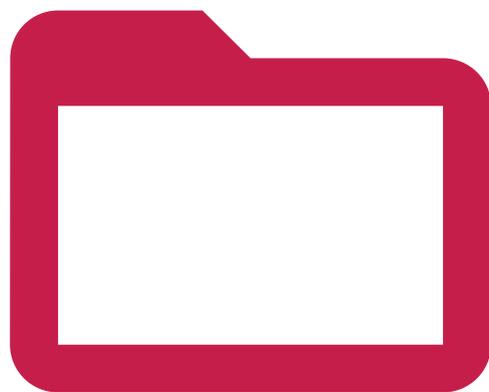
Par



RaspberryTips

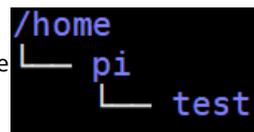
GESTION DES FICHIERS

Voici les commandes de base que tout débutant sous Linux doit apprendre afin de naviguer dans l'arborescence de fichiers



Note:

L'organisation des fichiers sous Linux est un arbre, dont la racine est /
Chaque nouveau sous-dossier ajoute un niveau en dessous de /



Par exemple, vous pouvez voir sur l'image ci-contre la représentation de la branche /home/pi/test

• CD <DOSSIER>

Changer de dossier, pour aller dans l'emplacement spécifié

Chemin absolu: `cd /home/pi/test`
Chemin relatif: `cd test`



NB: "Absolu" correspond au chemin en entier, alors que "Relatif" prend en compte le chemin à partir du chemin actuel (le second exemple ne marche que si vous êtes déjà dans /home/pi par exemple)

• MKDIR <DOSSIER>

Créer un nouveau dossier au nom indiqué (dans le répertoire courant ou spécifié)

Dossier en cours: `mkdir test`
Dossier spécifié: `mkdir /home/pi/test`



NB: Le premier exemple crée un dossier à l'emplacement actuel (chemin relatif)
Le second crée un dossier à l'emplacement spécifié (chemin absolu)

• MV <SRC> <DEST>

Déplace un fichier vers l'emplacement spécifié (permet aussi de renommer)

Un fichier: `mv test.txt /home/pi`
Un dossier: `mv /home/pi/test /home/pi/test2`



NB: La commande mv est forcément en mode récursif

• MORE <FICHIER>

Affiche le contenu d'un fichier, page par page

Chemin absolu: `more test.txt`
Chemin relatif: `more /home/pi/test.txt`



NB: Dans les longs fichiers, il faut appuyer sur "Espace" pour passer à la page suivante.

• LS (DOSSIER)

Lister les fichiers et dossiers présents dans l'emplacement indiqué

Dossier en cours: `ls`
Dossier spécifié: `ls /home/pi/test`



NB: Il existe plein d'options facultatives permettant de changer l'affichage du résultat, par exemple: `ls -latr /home/pi`

• CP <SOURCE> <DEST>

Copie un fichier d'un emplacement à un autre

Copier un fichier: `cp test.txt /home/pi`
Copier récursif: `cp -r /home/pi/test /home/user/`



NB: Utilisez l'option récursive afin de copier un dossier et tout son contenu

• CAT <FICHIER>

Affiche le contenu d'un fichier, sans pagination

Usage basique: `cat test.txt`
Avec un jocker: `cat *.txt`



NB: Le jocker vous permet d'afficher tous les fichiers respectant ces critères.

• TAIL <FICHIER>

Affiche les dernières lignes d'un fichier

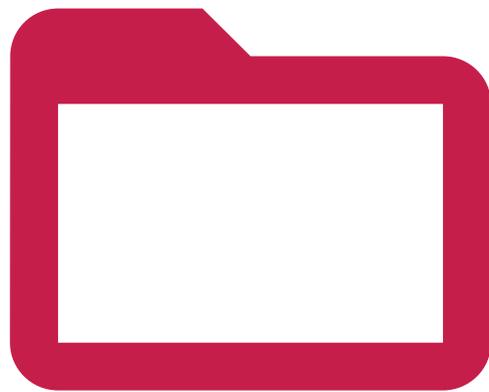
Base: `tail test.txt`
Lignes à prendre: `tail -n20 test.txt`
Temps réel: `tail -f test.txt`



NB: L'option -n permet de spécifier le nombre de lignes à afficher.
L'option -t permet d'afficher les nouvelles lignes au fur à mesure de leur écriture (parfait pour débbuger).



GESTION DES FICHIERS (2)



• HEAD <FICHIER>

Similaire à la commande tail mais pour afficher le début d'un fichier

10 lignes: `head test.txt`
20 lignes: `head -n20 test.txt`

• GREP

Grep est un outil puissant (mais complexe), permettant de faire des recherches:

Rechercher valeur: `grep "dhcp" /var/log/syslog`
Filtrer les résultats: `ls -latr | grep ".php"`
Ou sur un script: `/home/pi/script.sh | grep error`

 NB: Le pipe (|) vous permet d'exécuter une commande sur le résultat de la première.

• NANO <FICHIER>

Ouvre un fichier pour modification. Nano est un éditeur de fichier très puissant:

Usage basique: `nano /home/pi/test.txt`

 NB: Nano va créer le fichier s'il n'existe pas

• TAR

Tar est le gestionnaire de fichiers compressés sous Linux:

Créer un fichier: `tar -cvfz archive.tar.gz /home/pi/test`
Extraire le contenu: `tar -xvfz archive.tar.gz`

 Options:
-c pour Compresser, -x pour eXtraire
-v: mode verbeux, -z: utilise gZip pour la compression, -f spécifie le nom du fichier.
Utilisez "man tar" pour en savoir plus

• TOUCH <FICHIER>

Créé un nouveau fichier (vide)

Dossier courant: `touch test.txt`
Spécifique: `touch /home/pi/test.txt`

 NB: La plupart du temps, on utilisera plutôt "nano", car il permet de modifier le fichier directement

De nombreuses options avancées sont disponibles:

Exp. régulières: `grep "dhcp|dns" /var/log/syslog`
Options diverses: `grep -A2 -B4 'Fatal error' /var/log/syslog`
Recherche inversée: `grep -v 'Notice' /var/log/syslog`

Le symbole | dans les expressions régulières est un OU.
L'option -A permet d'afficher les lignes suivant le résultat, l'option -B permet la même chose avant le résultat.
L'option -v permet d'ignorer les lignes contenant la chaîne indiquée

• RM <FICHIER>

Supprimer un fichier ou un dossier:

Un fichier: `rm test.txt`
Un dossier: `rm -rf /home/pi/test`

 NB: L'option -rf sur un dossier permet de supprimer un dossier et tout son contenu (même s'il n'est pas vide).

• ZIP / UNZIP

Zip est similaire à tar, mais pour le format Zip (commun sous Windows):

Créer un fichier: `zip -r archive.zip /home/pi/test`
Extraire le contenu: `unzip archive.zip`

 NB: L'option -r permet de compresser tout le contenu du dossier.
L'option -d permet d'extraire à un emplacement spécifique.
Regardez "man zip" pour voir toutes les options disponibles.



GESTION DES FICHIERS (3)



- **PWD**

Une commande simple pour afficher l'emplacement actuel:

Exemple: `pwd`

- **FIND**

Find permet de faire une recherche sur votre Raspberry Pi:

Trouver un fichier: `find /home/pi -iname test.txt`
Filtrer les extensions: `find /home/pi -iname *.php`
Se limiter aux dossiers: `find / -type d -iname test`



NB: -iname est insensible à la casse (contrairement à -name). Vous pouvez utiliser "-type f" pour chercher uniquement les fichiers.

- **TREE**

Un autre outil permet d'afficher l'emplacement actuel, sous forme d'arbre:

Dossier courant: `tree`
Dossier spécifié: `tree /home/pi/`



NB: Il existe quelques options permettant de changer l'affichage du résultat, utilisez "man tree" pour en savoir plus.

Des options plus poussées:

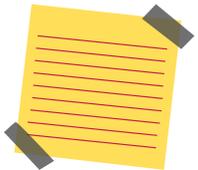
Taille des fichiers: `find / -size +10M`
Date modification: `find /home -mtime -2`
Lancer une commande sur les résultats:
`find /var/log -iname *.log.gz -exec rm {} \;`

La première commande affiche tous les fichiers de plus de 10M. L'option -mtime -2 liste les fichiers modifiés depuis 48h. Le paramètre {} dans la dernière commande sera remplacé par les résultats. Consultez "man find" pour plus de détails.



COMMANDES RESEAU

Voici toutes les commandes importantes concernant la gestion du réseau sur Raspberry Pi.



Note:

Les modèles récents de Raspberry Pi incluent deux interfaces: câble et Wi-Fi. L'interface Ethernet (câble) est nommé eth0, tandis que celle pour le Wi-Fi est wlan0.

● IFCONFIG

Affiche votre configuration réseau actuelle (adresse ip notamment):

Usage: `ifconfig`



NB: Il est possible d'ajouter le nom de l'interface en paramètre si vous voulez: "ifconfig wlan0".

● IFUP / IFDOWN

Permet d'activer ou de désactiver une interface:

Activation: `sudo ifup eth0`
Désactivation: `sudo ifdown eth0`



NB: Cela peut être utile pour désactiver le Wi-Fi une fois connecté en câble par exemple.

● HOSTNAME

Affiche ou change le nom d'hôte du Raspberry Pi:

Affichage: `hostname`
Configuration: `sudo hostname RaspberryZero`

● SSH <USER>@<IP>

Connexion à un autre système via SSH:

Exemple: `ssh pi@192.168.1.1`

● RSYNC

Similaire à scp mais avec davantage de fonctionnalités:

Syntaxe: `rsync <file> <user>@<ip>:<path>`
Exemple: `rsync test.txt pi@192.168.1.1:/home/pi/`



NB: Utilisez "man rsync" pour toutes les options

● IWCONFIG

Montre les informations concernant la connexion Wi-Fi (SSID, vitesse, ...):

Usage: `iwconfig`



NB: Vous pouvez également spécifier l'interface: iwconfig wlan0

● PING <HOTE>

Vérifier si une machine est joignable sur le réseau:

Usage: `ping 192.168.1.1`



NB: Consultez l'aide "man ping" pour toutes les options.

● WGET <URL>

Télécharge un fichier depuis la console:

Usage: `wget http://192.168.1.1/test.txt`
Spécifier le nom de fichier:
`wget http://192.168.1.1/test.txt -O target.txt`

● SCP

Copie un fichier sur le réseau à travers SSH:

Syntaxe: `scp <file> <user>@<ip>:<path>`
Exemple: `scp test.txt pi@192.168.1.1:/home/pi/`

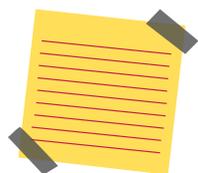
Copie locale: `rsync /home/pi/* /media/usb/`
Copie distante récursive: `rsync -auzr /home/pi/Documents/* pi@192.168.1.1:/home/pi/Documents/`



GESTION DES PAQUETS



Une fois le réseau configuré, vous allez passer à la mise à jour du système et à l'installation de nouveaux programmes. Cette page vous donne toutes les commandes pour le faire depuis le terminal.



Vocabulaire:

Sous Linux, chaque programme est un **paquet**, tout comme chaque **dépendance**. Vous téléchargez les paquets depuis un **dépôt** (serveur hébergeant les paquets). Vous aurez besoin d'un outil appelé **apt** pour rechercher, installer et mettre à jour les paquets sur Debian/Raspbian. Toutes ces commandes nécessitent les droits administrateur, il faudra donc utiliser **sudo**.

• APT UPDATE

Télécharge la liste des paquets à jour depuis le dépôt.

Usage: `sudo apt update`



NB: Pour ajouter ou modifier les dépôts utilisés, vous pouvez éditer le fichier `/etc/apt/sources.list` et suivre les instructions.

• APT UPGRADE

Télécharge et installe les dernières mises à jour pour les paquets utilisés:

Usage: `sudo apt upgrade`



NB: L'utilisation de "apt update" avant est obligatoire. L'option `-y` permet d'accepter automatiquement les mises à jour (sans confirmation).

• RPI-UPDATE

Met tout à jour sur votre Raspberry Pi. À utiliser en dernier recours:

Usage: `sudo rpi-update`

• APT INSTALL <PAQUET>

Installe le paquet spécifié sur votre système:

Usage: `sudo apt install phpmyadmin`



NB: Utilisez la commande "search" ci-dessous pour trouver le nom exact d'un paquet.

• APT REMOVE <PAQUET>

Désinstalle un paquet du système:

Usage: `sudo apt remove vim`



NB: Je vous donne la commande permettant de lister les paquets installés juste après.

• APT SEARCH

Très utile pour retrouver le nom d'un paquet:

Usage: `apt search openjdk`
Avec *grep*: `apt search openjdk | grep jre`



NB: L'utilisation de `sudo` n'est pas utile pour cette commande.

• INSTALLATION MANUELLE

Parfois, vous devrez installer des paquets à la main, voici comment faire:

Télécharger le paquet avec *wget*:

`wget https://www.realvnc.com/download/file/viewer.files/VNC-Viewer-6.19.325-Linux-ARM.deb`

Installez-le à la main:

`sudo dpkg -i VNC-Viewer-6.19.325-Linux-ARM.deb`



NB: La commande `dpkg -r` permet de désinstaller un paquet s'il a été installé à la main.

• LISTE DES PAQUETS

`Dpkg` permet aussi d'afficher la liste des paquets installés sur votre système:

Syntaxe: `dpkg -l`
Avec *grep*: `dpkg -l | grep php`

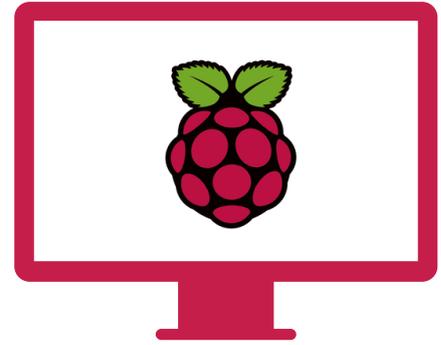


NB: Lisez l'aide "man dpkg" pour connaître toutes les options disponibles.



GESTION DU SYSTÈME

Maintenant que vos programmes sont installés, nous pouvons passer à des commandes plus avancées sur la gestion du système.



• REBOOT

Cette commande redémarre instantanément le Raspberry Pi:

Usage: `sudo reboot`

• SERVICE

Permet de gérer le fonctionnement des services du système:

Démarrage: `sudo service apache2 start`
Arrêt: `sudo service apache2 stop`
Redémarrer: `sudo service apache2 start`
Recharger: `sudo service apache2 reload`



NB: Utilisez "service <service>" pour connaître toutes les options disponibles. La touche "TAB" vous aidera à trouver le nom du service.

• LISTE DES PROCESSUS

Affiche tous les processus en cours:

Usage basique: `ps aux`
Filtrer par utilisateur: `ps -u pi`

• HTOP

Une bonne alternative à "top" pour afficher l'état du système:

Usage: `htop`



NB: Si htop n'est pas disponible, c'est qu'il faut d'abord l'installer: "sudo apt install htop".

• SHUTDOWN

Arrête le Raspberry Pi, soit immédiatement, soit à une heure spécifiée:

Maintenant: `sudo shutdown -h now`
Heure indiquée: `sudo shutdown -h 20:00`

• DEMARRAGE AUTO

Permet de gérer le démarrage automatique ou non lors du lancement du Raspberry Pi:

Démarrage auto: `sudo update-rc.d ssh enable`
Désactivation: `sudo update-rc.d -f ssh remove`



NB: Pour lancer un script au démarrage, ajoutez le dans /etc/rc.local

• KILL / KILLALL

Termine immédiatement le processus indiqué:

Kill: `kill 12345`
Killall: `killall php`



NB: Utilisez la commande "ps aux" au préalable pour trouver le numéro du processus.

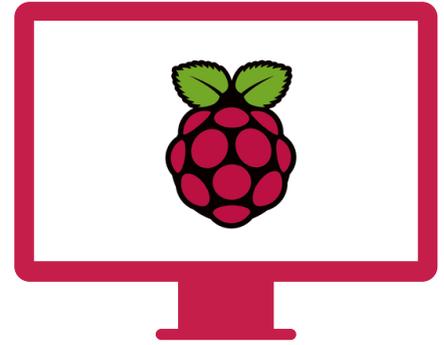
• DF

Affiche la liste des partitions, ainsi que l'espace disque utilisé/restant:

Usage basique: `df`
Plus lisible: `df -h`
Partition spécifiée: `df -h /media/usb`



GESTION DU SYSTÈME (2)



• DU

Affiche l'espace disque utilisé par le dossier courant ou spécifié:

Usage basique: `du`
Dossier précis: `du /home/pi`
Résumé: `du --summarize /home/pi`
Plus gros fichiers: `du -ak | sort -nr | head -20`



NB: Il existe bien plus d'options disponibles, utilisez "man du" pour en savoir plus.

• DATE

Comme le nom l'indique, permet d'afficher la date:

Affichage complet: `date`
Format précis: `date +%m-%d-%Y`



NB: Le "man date" vous aidera au niveau du format notamment

• CHOWN

Changer le propriétaire d'un fichier :

Propriétaire: `sudo chown pi /usr/local/bin/script.sh`
Propriétaire et groupe: `sudo chown pi:www-data /var/www/html/my-site`

• TEMPERATURE DU CPU

Elle n'est pas facile à se souvenir, mais elle peut être très utile en cas de surcharge:

Usage: `vcgencmd measure_temp`



NB: vcgencmd est inclus dans le paquet libraspberrypi-bin, une installation peut être nécessaire: "sudo apt install libraspberrypi-bin"

• MOUNT

Monter une partition (une clé USB par exemple)

Montage: `sudo mount /dev/sda1 /mnt/usb`
Démontage: `sudo umount /mnt/usb`



NB: Ce n'est pas très simple pour les débutants, je vous conseille la lecture de ce tutoriel si vous n'avez jamais fait: <https://raspberrytips.fr/monter-cle-usb-raspberry-pi/>

• UPTIME

Affiche le temps d'exécution du Raspberry Pi depuis le dernier démarrage:

Utilisation: `uptime`
Heure dernier boot: `uptime -s`

• CHMOD

Change les permissions d'un fichier ou dossier:

Format numérique: `chmod 644 script.sh`
Format lettres: `chmod +x script.sh`



NB: Il existe des outils pour vous aider à connaître les bonnes options à utiliser, par exemple: <https://chmod-calculator.com/>

• MAN <COMMANDE>

Je vous l'ai déjà donné plein de fois dans ce guide, permet d'afficher l'aide:

Exemple: `man find`



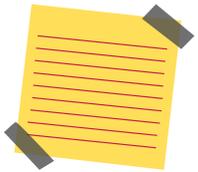
NB: Utilisez la touche "Espace" pour passer à la page suivante, et "q" pour quitter.



COMMANDES RASPBIAN



La plupart des commandes présentées sont des commandes Debian, mais sur cette page je vous donne les commandes spécifiques à Raspbian.



Note:

Il existe quelques commandes qui n'existent que sur Raspbian. Elles ne sont pas forcément essentielles pour l'utilisation d'un Raspberry Pi. Mais sur la plupart des sites, elles ne vous seront pas données car spécifiques à notre système favori :)

● RASPI-CONFIG

C'est l'outil principal pour la configuration d'un Raspberry Pi:

Usage: `sudo raspi-config`



NB: Raspi-config vous permet de changer plein de paramètres systèmes via un menu pratique (localisation, Wi-Fi, interfaces, etc.)

● RASPIVID

Même chose, mais pour prendre une vidéo avec la caméra:

Utilisation: `raspivid -o video.h264 -t 10000`



NB: L'option `-t` permet de spécifier la durée du film

● RASPIVIDYUV / RASPIYUV

Idem que raspivid mais au format YUV:

Utilisation: `raspividuyv -o video.yuv`

● RASPISTILL

Cette commande vous permet de prendre une photo avec la caméra du Raspberry Pi:

Utilisation: `raspistill -o image.jpg`

● RASPI-GPIO

Permet d'accéder aux pins GPIO directement sans code:

Récupérer valeur: `raspi-gpio get`

Définir valeur: `raspi-gpio set 20 a5`

● RPI-UPDATE

Nous avons déjà vu cette commande (mise à jour complète du système):

Usage: `sudo rpi-update`



COMMANDES DIVERSES

Dans cette page, je vous donne toutes les commandes utiles, mais qui ne rentrent pas dans les autres catégories :)



● HISTORY

Linux archive toutes les commandes tapées, la commande history permet de les afficher:

Voir tout: `history`
Les 20 dernières: `history | tail -n 20`
Tout effacer: `history -c`
Effacer une ligne: `history -d 123`

● |

Je vous ai déjà montré ce symbole précédemment, mais il est important:

Syntaxe: `<command1> | <command2>`
Grep: `cat test.txt | grep error`
Double: `du -ak | sort -nr | head -20`

● !

Lance une commande présente dans l'historique:

Syntaxe: `!<history_id>`
Exemple: `!123`



NB: Pensez bien à vérifier le numéro à chaque fois, il change à chaque nouvelle commande tapée :)

● >

Stocke le résultat dans un fichier:

Syntaxe: `<command> > <filename>`
Exemple: `cat test.txt | grep error > error.log`



NB: La dernière commande stock toutes les lignes contenant le mot clé "error" dans le fichier error.log. Cette commande n'affichera rien du tout.

● CRONTAB

Planificateur de tâches pour Raspberry Pi:

Voir les tâches: `crontab -l`
Modifier: `crontab -e`



NB: La syntaxe n'est pas évidente à maîtriser, je vous conseille de la valider avec ce genre d'outil: <https://crontab.guru/>

● SCREEN

Lance un terminal virtuel, permettant de laisser tourner un script en arrière plan:

Lancement: `screen -S <name>`
Quitter: `CTRL+A CTRL+D`
Revenir: `screen -r <name>`
Arrêter: `CTRL+D`

● !!

Similaire à ! mais pour la dernière commande:

Usage: `!!`



NB: Peut être pratique pour lancer la même commande plusieurs fois.

● >>

Même chose mais ajoute à la fin du fichier au lieu de remplacer tout:

Usage: `cat test.txt | grep error >> error.log`



NB: Dans ce cas, les lignes d'erreur sont ajoutées à la fin de error.log.



COMMANDES D'EXPERT

Et pour finir, voyons quelques commandes bien compliquées qui devraient ravir ceux qui savaient tout jusqu'à présent :) Bien que très utiles, elles ne sont pas accessibles à tous.



● AWK

Awk est pratiquement un langage de programmation. Il permet de rechercher une valeur et de transformer l'affichage du résultat:

Syntaxe: `awk [-F] [-v var=value] 'program' file`
Exemple basique: `awk -F":" '{print $1}' /etc/passwd`



NB: La dernière commande affiche juste la première colonne.
Difficile d'expliquer en deux lignes, je vous recommande la lecture de ce guide: <https://do.co/2VC8mmn>

● CUT

Une troisième manière de transformer l'affichage d'un fichier:

Syntaxe: `cut <option> <file>`
Exemple: `cut -d : -f 1 /etc/passwd`



NB: -d indique le délimiteur, and -f le champ à garder. Utilisez "man cut" pour voir les autres options.

● LSOF

Abréviation de "LiSt Open Files", affiche les fichiers ouverts sur votre système:

Usage: `lsdf`



NB: À coupler avec grep pour filtrer les résultats

● NETSTAT

Surveille l'activité réseau:

Ports ouverts: `netstat -l`
Indique le processus: `netstat -lp`
Temps réel: `netstat -lpc`



NB: Il existe plein d'autres options, utilisez "man netstat" pour en savoir plus

● SED

Similaire à awk, mais utilisant uniquement des expressions régulières:

Syntaxe: `sed <option> <script> <file>`
Exemple: `sed '/^#/d' /etc/apache2/apache2.conf`



NB: La dernière commande supprime tous les commentaires dans le fichier de configuration Apache.
Comme pour awk, je vous conseille de lire des tutoriels avant de vous lancer.

● WC

WC est l'abréviation de "Words Count" et affiche le nombre de mots, lignes ou caractères:

Syntaxe: `wc <options> <file>`
Nb lignes: `wc -l /var/log/syslog`



NB: -l pour les lignes, -w pour les mots et -m pour les caractères.

● WATCH

Surveille le résultat d'une commande, à intervalles réguliers:

Utilisation: `watch date`
Fréquence : `watch -n10 date`



NB: Par défaut l'affichage est rafraîchi toutes les 2s

● DMSG

Affiche les logs du dernier démarrage du système:

Usage: `dmesg`



NB: A coupler avec grep pour filtrer les erreurs par exemple

Merci pour votre lecture!
À bientôt sur RaspberryTips
Patrick

