tutoriel

74 Commandes pour Raspberry Pi - Antisèche commandes Raspberry Pi

Gestion des fichiers

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

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, voici la représentation de la branche /home/pi/test

```
/home
└─pi
  └─test
```

1. **cd <DOSSIER>** : Changer de dossier, pour aller dans l'emplacement spécifié
   - Chemin absolu:

     ```
     ...@...:~ $ cd /home/pi/test
     ```

   - Chemin relatif:

     ```
     ...@...:~ $ cd test
     ```

   - “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)
2. **mkdir <DOSSIER>** : Créer un nouveau dossier au nom indiqué (dans le répertoire courant ou spécifié)
3. **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
```

4. **more <FICHIER>** : Affiche le contenu d'un fichier, page par page
   - Chemin absolu:

```
...@...:~ $ more test.txt
```

   - Chemin relatif:

```
...@...:~ $ more /home/pi/test.txt
```

Dans les longs fichiers, il faut appuyer sur “Espace” pourpasser à la page suivante.

5. **ls (DOSSIER)** : Lister les fichiers et dossiers présents dans l'emplacement indiqué
   - Dossier en cours:

```
...@...:~ $ ls
```

   - Dossier spécifié:

```
...@...:~ $ ls /home/pi/test
```

Il existe plein d'options facultatives permettant de changer l'affichage du résultat, par exemple:

```
...@...:~ $ ls -latr /home/pi
```

6. **cp <SOURCE> <DEST>** : Copie un fichier d'un emplacement à un autre
   1. Copier un fichier:

```
...@...:~ $ cp test.txt /home/pi
```

2. Copie récursive:

```
...@...:~ $ cp -r /home/pi/test /home/user/
```

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

7. **cat <FICHIER>** : Affiche le contenu d'un fichier, sans pagination
   - Usage basique:

```
...@...:~ $ cat test.txt
```

   - Avec un jocker:

```
...@...:~ $ cat *.txt
```

8. **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
```

9. **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
```

10. **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
```

- Le pipe (|) vous permet d'exécuter une commande sur le résultat de la première
- De nombreuses options avancées sontdisponibles:
  - 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/syslogi
```

11. **nano <FICHIER>** : Ouvre un fichier pour modification. Nano est un éditeur de fichier très puissant
    - Usage basique:

```
...@...:~ $ nano /home/pi/test.txt
```

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

12. **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

13. **touch <FICHIER>** : Crée un nouveau fichier (vide)
    - Dossier courant:

```
...@...:~ $ touch test.txt
```

    - Spécifique:

```
...@...:~ $ touch /home/pi/test.txt
```

    - La plupart du temps, on utilisera plutôt “nano”, car il permetde modifier le fichier directement
14. **rm <FICHIER>** : Supprimer un fichier ou un dossier
    - Un fichier:

```
...@...:~ $ rm test.txt
```

    - Un dossier:

```
...@...:~ $ rm -rf /home/pi/test
```

    - L'option -rf sur un dossier permet de supprimer un dossier et tout son contenu (même s'il n'est pas vide).
15. **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
```

    - L'option -r permet de compresser tout le contenu dudossier.
    - L'option -d permet d'extraire à un emplacement spécifique.
    - Regardez “man zip” pour voir toutes les options disponibles

16. **pwd** : Une commande simple pour afficher l'emplacement actuel
    - Exemple:pwd
17. **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
```

    - -iname est insensible à la casse (contrairement à -name).
    - Vous pouvez utiliser “-type f” pour chercher uniquement les fichiers
18. **tree** : Un autre outil permet d'afficher l'emplacement actuel, sous forme d'arbre
    - Dossier courant:

```
...@...:~ $ tree
```

    - Dossier spécifié:

```
...@...:~ $ tree /home/pi/
```

    - Il existe quelques options permettant de changer l'affichage du résultat, utilisez “man tree” pour en savoir plus
19. **Des options plus poussées**:
    - Taille des fichiers:

```
...@...:~ $ find / -size +10M
```

      affiche tous les fichiers de plus de 10M

    - Date modification:

```
...@...:~ $ find /home -mtime -2
```

      -mtime -2 liste les fichiers modifiés depuis 48h

    - Lancer une commande sur les résultats:

```
...@...:~ $ find /var/log -iname *.log.gz -exec rm {} \;
```

Le paramètre {} sera remplacé par les résultats

- Consultez “man find” pour plus de détails

Commandes réseau

1. **ifconfig** : Affiche votre configuration réseau actuelle(adresse ip notamment)
   - Usage:

```
...@...:~ $ ifconfig
```

   - Il est possible d'ajouter le nom de l'interface en paramètre si vous voulez: “ifconfig wlan0”.
2. **ifup / ifdown** : Permet d'activer ou de désactiver une interface
   - Activation:

```
...@...:~ $ sudo ifup eth0
```

   - Désactivation:

```
...@...:~ $ sudo ifdown eth0
```

   - Cela peut être utile pour déactiver le Wi-Fi une fois connecté en câble par exemple.
3. **hostname** : Affiche ou change le nom d'hôte du Raspberry Pi:
   - Affichage:

```
...@...:~ $ hostname
```

   - Configuration:

```
...@...:~ $ sudo hostname RaspberryZero
```

4. **ssh <USER>@<IP>** : Connexion à un autre système via SSH
   - Exemple:ssh pi@192.168.1.1
5. **rsync** : Similaire à scp mais avec davantage defonctionnalités
   - Syntaxe:

```
...@...:~ $ rsync <file> <user>@<ip>:<path>
```

- Exemple:

```
...@...:~ $ rsync 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/
```

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

```
...@...:~ $ iwconfig
```

   - NB: Vous pouvez également spécifier l'interface:

```
...@...:~ $ iwconfig wlan0
```

7. **ping <HOTE>** : Vérifier si une machine est joignable sur le réseau
   - Usage:

```
...@...:~ $ ping 192.168.1.1
```

   - Consultez l'aide “man ping” pour toutes les options
8. **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 -Otarget.txt
```

9. **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/
```

Gestion des paquets

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 de l'outil **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**.

1. **sudo apt update** : Télécharge la liste des paquets à jour depuis le dépôt
   - Usage :

```
...@...:~ $ sudo apt update
```

   - Pour ajouter ou modifier les dépôts utilisés, vous pouvez éditer le fichier /etc/apt/sources.list et suivre les instructions
2. **sudo rpi-update** : Met tout à jour sur votre Raspberry Pi. À utiliser en dernier recours
   - Usage :

```
...@...:~ $ sudo rpi-update
```

3. **sudo apt remove <PAQUET>** : Désinstalle un paquet du système
   - Usage :

```
...@...:~ $ sudo apt remove vim
```

   - Je vous donne la commande permettant de lister les paquets installés juste après
4. **Installation manuelle** : Parfois, vous devrez installer des paquets à la main,voici comment faire :
   1. **Téléchargez le paquet** avec wget :

```
...@...:~ $ wget
https://www.realvnc.com/download/file/viewer.files/VNC-Viewer-6.19
.325-Linux-ARM.deb
```

   2. **Installez-le à la main** :

```
...@...:~ $ sudo dpkg -i VNC-Viewer-6.19.325-Linux-ARM.deb
```

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

5. **sudo apt upgrade** : Télécharge et installe les dernières mises à jour pour les paquets utilisés
   - Usage :

```
...@...:~ $ sudo apt upgrade
```

   - L'utilisation de “apt update” avant est obligatoire. L'option -y permet d'accepter automatiquement les mises à jour (sans confirmation)
6. **sudo apt install <paquet>** : Installe le paquet spécifié sur votre système
   - Usage :

```
...@...:~ $ sudo apt install phpmyadmin
```

   - Utilisez la commande “search” ci-dessous pour trouver le nom exact d'un paquet
7. **sudo apt search** : Très utile pour retrouver le nom d'un paquet
   - Usage :

```
...@...:~ $ apt search openjdk
```

   - Avec grep :

```
...@...:~ $ apt search openjdk | grep jre
```

   - L'utilisation de sudo est inutile pour cette commande
8. **Liste des paquets installés** : Dpkg permet aussi d'afficher la liste des paquets installés sur votre système
   - Syntaxe :

```
...@...:~ $ sudo dpkg -l
```

   - Avec grep :

```
...@...:~ $ sudo dpkg -l | grep php
```

   - Lisez l'aide “man dpkg” pour connaître toutes les optionsdisponibles

Gestion du système

1. **reboot** : Cette commande redémarre instantanément le Raspberry Pi
    - Usage :

```
...@...:~ $ sudo reboot
```

2. **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
```

    - Utilisez “service <service>” pour connaitre toutes les options disponibles.
    - La touche TAB vous aidera à trouver le nom du service
3. **Liste des processus** : Affiche tous les processus en cours
    - Usage basique :

```
...@...:~ $ ps aux
```

    - Filtrer par utilisateur :

```
...@...:~ $ ps -u pi
```

4. **htop** : Une bonne alternative à “top” pour afficher l'état du système
    - Usage :

```
...@...:~ $ htop
```

- Si htop n'est pas disponible, c'est qu'il faut d'abord l'installer:

```
...@...:~ $ sudo apt install htop
```

5. **shutdown** : Arrête le Raspberry Pi, soit immédiatement, soit à une heure spécifiée
   - Maintenant:

```
...@...:~ $ sudo shutdown -h now
```

   - Heure indiquée:

```
...@...:~ $ sudo shutdown -h 20:00
```

6. **Démarrage 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
```

   - Pour lancer un script au démarrage, ajoutez le dans /etc/rc.local
7. **kill / killall** : Termine immédiatement le processus indiqué
   - Kill :

```
...@...:~ $ kill 12345
```

   - Killall :

```
...@...:~ $ killall php
```

   - Utilisez la commande “ps aux” au préalable pour trouver le numéro du processus
8. **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
```

9. **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
```

   - Il existe bien plus d'options disponibles, utilisez “man du” pour en savoir plus
10. **date** : Comme le nom l'indique, permet d'afficher la date
   - Affichage complet :

```
...@...:~ $ date
```

   - Format précis :

```
...@...:~ $ date +%m-%d-%Y
```

   - Le “man date” vous aidera au niveau du format notamment
11. **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/mysite
```

12. **Température du cpu** : Elle n'est pas facile à se souvenir, mais elle peut être très utile en cas de surcharge
    - Usage :

```
...@...:~ $ vcgencmd measure_temp
```

    - vcgencmd est inclus dans le paquet libraspberry-bin, une installation peut être nécessaire:

```
...@...:~ $ sudo apt install libraspberrypi-bin
```

13. **mount** : Monter une partition (une clé USB par exemple)
    - Montage :

```
...@...:~ $ sudo mount /dev/sda1 /mnt/usb
```

    - Démontage :

```
...@...:~ $ sudo umount /mnt/usb
```

    - 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/
14. **uptime** : Affiche le temps d'exécution du Raspberry Pi depuis le dernier démarrage
    - Utilisation :

```
...@...:~ $ uptime
```

    - Heure dernier boot :

```
...@...:~ $ uptime -s
```

15. **chmod** : Change les permissions d'un fichier ou dossier
    - Format numérique :

```
...@...:~ $ chmod 644 script.sh
```

    - Format lettres :

```
...@...:~ $ chmod +x script.sh
```

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

16. **man <COMMANDE>** : permet d'afficher l'aide
    - Utilisez la touche Espace pour passer à la page suivante et Q pour quitter.
    - Exemple :

```
...@...:~ $ man find
```

Commandes Raspbian

1. **raspi-config** : C'est l'outil principal pour la configuration d'un Raspberry Pi
    - Usage :

```
...@...:~ $ sudo raspi-config
```

    - Raspi-config permet de changer plein de paramètres système via un menu pratique (localisation, Wi-Fi, interfaces,etc.)
2. **raspivid** : Même chose, mais pour prendre une vidéo avec la caméra
    - Utilisation :

```
...@...:~ $ raspivid -o video.h264 -t 10000
```

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

3. **raspividyuv / raspiyuv** : Idem que raspivid mais au format YUV
    - Utilisation :

```
...@...:~ $ raspividyuv -o video.yuv
```

4. **raspistill** : Cette commande permet de prendre une photo avec la caméra du Raspberry Pi
    - Utilisation :

```
...@...:~ $ raspistill -o image.jpg
```

5. **raspi-gpio** : Permet d'accéder aux pins GPIO directement sans code
    - Récupérer valeur :

```
...@...:~ $ raspi-gpio get
```

    - Définir valeur :

```
...@...:~ $ raspi-gpio set 20 a5
```

6. **rpi-update** : mise àjour complète du système
    - Usage :

```
...@...:~ $ sudo rpi-update
```

Commandes diverses

1. **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
```

2. **|** : ce symbole est important
    - Syntaxe :

```
...@...:~ $ <command1> | <command2>
```

    - Grep :

```
...@...:~ $ cat test.txt | grep error
```

    - Double:

```
...@...:~ $ du -ak | sort -nr | head -20
```

3. **!** : Lance une commande présente dansl'historique
    - Syntaxe :!

```
...@...:~ $ <history_id>
```

    - Exemple :

```
...@...:~ $ !123
```

    - Pensez bien à vérifier le numéro à chaque fois, il change à chaque nouvelle commande tapée
4. **>** : Stocke le résultat dans un fichier
    - Syntaxe :

```
...@...:~ $ <command> > <filename>
```

    - Exemple :

```
...@...:~ $ cat test.txt | grep error > error.log
```

    - La dernière commande stocke toutes les lignes contenant le mot clé “error” dans le fichier error.log. Cette commande n'affichera rien du tout
5. **crontab** : Planificateur de tâches pour Raspberry Pi
    - Voir les tâches :

```
...@...:~ $ crontab -l
```

    - Modifier :

```
...@...:~ $ crontab -e
```

    - La syntaxe n'est pas évidente à maitriser, je vous conseille de la valider avec ce genre d'outil :https://crontab.guru/
6. **!!** : Similaire à ! mais pour la dernière commande
    - Usage :

```
...@...:~ $ !!
```

7. **»** : Même chose que > mais ajoute à la fin du fichier au lieu de remplacer tout
    - Usage :

```
...@...:~ $ cat test.txt | grep error >> error.log
```

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

Commandes d'expert

1. **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
```

    - 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/2VC8mnm
2. **cut** : Une troisième manière de transformer l'affichage d'un fichier
    - Syntaxe :

```
...@...:~ $ cut <option> <file>
```

    - Exemple :

```
...@...:~ $ cut -d : -f 1 /etc/passwd
```

    - -d indique le délimiteur, and -f le champ à garder.
    - Utilisez “man cut” pour voir les autres options
3. **lsof** : Abréviation de “LiSt Open Files”, affiche les fichiers ouverts sur votre système
    - Usage :

```
...@...:~ $ lsof
```

    - À coupler avec grep pour filtrer les résultats
4. **netstat** : Surveille l'activité réseau
    - Ports ouverts :

```
...@...:~ $ netstat -l
```

    - Indique le processus :

```
...@...:~ $ netstat -lp
```

- Temps réel :

```
...@...:~ $ netstat -lpc
```

- Il existe plein d'autres options, utilisez “man netstat” pour en savoir plus

5. **sed** : Similaire à awk, mais utilisant uniquement des expressions régulières
   - Syntaxe :

```
...@...:~ $ sed <option> <script> <file>
```

   - Exemple :

```
...@...:~ $ sed '/^#/d' /etc/apache2/apache2.conf
```

   - La dernière commande supprime tous les commentaires dans le fichier de configuration Apache.
   - Comme pour awk, je vous conseille de lire des tutorials avant de vous lancer
6. **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
```

   - -l pour les lignes, -w pour les mots et -m pour les caractères
7. **watch** : Surveille le résultat d'une commande, à intervalles réguliers
   - Utilisation :

```
...@...:~ $ watch date
```

   - Fréquence :

```
...@...:~ $ watch -n10 date
```

   - Par défaut l'affichage est rafraichi toutes les 2s
8. **dmesg** : Affiche les logs du dernier démarrage du système

- Usage :

```
...@...:~ $ dmesg
```

- A coupler avec grep pour filtrer les erreurs par exemple

Voir aussi

- **(fr)** http://Article

Basé sur « 74 Commandes pour Raspberry Pi » par raspberrytips.fr.

From:
https://doc.nfrappe.fr/ - **Documentation du Dr Nicolas Frappé**

Permanent link:
https://doc.nfrappe.fr/doku.php?id=tutoriel:nanopc:raspi:74commandes:start

Last update: **2022/11/08 19:40**