

# Commandes utiles avec Linux

## Obtenir l'explication d'une commande

copiez/collez une commande sur <https://explainshell.com/> et vous aurez une explication de sa syntaxe et de son effet.

## Envoyer le résultat d'une commande en ligne

```
ls | nc termbin.com 9999
```

## Systeme

### Rendre un fichier exécutable

```
chmod +x script.sh # Rend le fichier script.sh executable
```

### Caractéristiques du système

```
inxi -Fxxrz  
inxi -ACMSxxz https://smxi.org/docs/inxi-options.htm
```

### Caractéristiques processeur

```
cat /proc/cpuinfo
```

### Taille de la RAM

```
grep MemTotal /proc/meminfo
```

### Type de linux

```
uname -srm
```

### Type de système d'affichage (X11, Wayland)

```
echo $XDG_SESSION_TYPE
```

### Version d'OS

```
lsb_release -a
```

### Connaître l'environnement de bureau

```
echo "${XDG_CURRENT_DESKTOP}"
```

## Espace disque

```
df -h
```

## Version OpenGL

```
sudo apt install mesa-utils  
glxinfo | grep "OpenGL version"
```

## Quelle version de GCC ?

GCC : GNU C Compiler, comprend différents compilateurs : C (gcc), C++ (g++), etc.

```
gcc -v ou g++ -v
```

```
ldd --version # pour trouver la version de glibc
```

## Localiser un exécutable

```
which nom_de_la_commande
```

## Changer date et heure

```
sudo date +%T -s "10:13:13"  
sudo date +%Y%m%d -s "20081128"
```

## Customisation

### Installer une police de caractères (avec Debian)

Copier les fichiers de la fonte (.ttf ou .otf) dans :

```
/usr/local/share/fonts (system-wide)  
~/.local/share/fonts (user-specific)  
~/.fonts (user-specific)
```

Vérifier que les fichiers ont bien les droits 644 (-rw-r-r-)

```
fc-cache -v # pour mettre à jour les cache des fontes  
fc-list # pour vérifier
```

On peut aussi utiliser accessoires/font-manager ou fontconfig (cf. <https://wiki.debian.org/Fonts> )

## Écran

### Changer la luminosité de l'écran

```
xrandr|grep ' connected '|awk '{print $1}' # renvoie le nom de l'écran, par ex. VGA-1  
# ou, variante : xrandr -q |grep " connected"  
xrandr --output VGA-1 --brightness 0.5
```

## Stockage (Disques/Dossiers/Fichiers)

### Clé USB en read-only

```
mount # pour voir le statut des clés (ro : read-only, rw : read-write)  
sudo mount -o remount,rw /media/$USER/*
```

Une autre possibilité

```
df -h // repérer le nom du volume correspondant à la carte SD  
sudo umount /dev/sdx1 // démonter le volume au préalable
```

```
sudo fsck -av /dev/sdx1 // réparer le volume
```

## Liste de fichiers

La commande `tree` est très pratique pour afficher une arborescence ou rechercher certains fichiers dans une arborescence, quelques exemples

```
sudo apt install tree
tree ./sketchbook/2013B           # fichiers, sous-dossiers à partir du chemin donné
tree -a ./sketchbook/2013B       # idem, en affichant les fichiers/dossiers cachés
tree --du ./sketchbook/2013B     # afficher la taille des sous-dossiers
tree --du ./sketchbook/2013B -o liste.txt # envoyer le résultat de la commande dans un fichier texte
```

## Copie/déplacement de fichiers/dossiers

### Copie

la commande `cp` permet de copier un fichier (`cp -r` pour un dossier)

```
cp /un/fichier /un/autre/fichierAvecUnNouveauNom
cp /un/fichier /un/dossier/
cp -r /un/dossier /un/autre/
```

### Raccourcis

```
ln -s /chemin/vers/dossier /home/user/Desktop/dossier_raccourci
```

### Déplacement

## Espace disque

### Afficher l'espace libre

```
df -h
```

Sys. de fichiers	Taille	Utilisé	Dispo	Uti%	Monté sur
dev	7,7G	0	7,7G	0%	/dev
run	7,7G	2,0M	7,7G	1%	/run
/dev/nvme0n1p2	922G	457G	418G	53%	/
tmpfs	7,7G	314M	7,4G	5%	/dev/shm
tmpfs	7,7G	61M	7,6G	1%	/tmp
/dev/nvme0n1p1	300M	312K	300M	1%	/boot/efi
tmpfs	1,6G	76K	1,6G	1%	/run/user/1000

### Afficher la taille d'un dossier

```
du -sh ~/tmp/
```

```
17M    /home/jumbef/tmp/
```

### Afficher l'utilisation de l'espace disque

Sous forme d'une arborescence dans laquelle on peut se déplacer interactivement. La commande démarre à partir du répertoire d'où elle est exécutée.

```
ncdu # pour "NCurses Disk Usage, q pour quitter ncdu
```

```
ncdu -x / # afficher uniquement la partition racine aka slash
```

## Réseau

### Quelle est l'adresse IP de la machine ?

```
hostname -I
```

## Quelle est l'adresse IP de la passerelle ?

```
ip r
```

## Quel est le masque de sous-réseau ?

En clair

```
sudo ifconfig
```

Le masque de sous-réseau est aussi affiché par les commandes `ip a` et autres en [notation CIDR](#)

## Quelle machine est visible sur le réseau local ?

À adapter en fonction de l'adresse IP de la passerelle/routeur

```
sudo nmap -sP 192.168.1.*
```

## Trouver une IP libre sur le réseau

(Pour la donner à un ESP32)

```
sudo arp-scan -I wlp3s0 192.168.1.1/24
```

OU

```
sudo nmap -sP -PR 192.168.1.* # (plus complet)
```

## Afficher l'état des connexions réseaux (dont les adresses IP)

`ip addr` (Debian >= 8 | Ubuntu >= 14.04 )

```
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default
   link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
   inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
   inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
   link/ether 08:00:27:4f:41:81 brd ff:ff:ff:ff:ff:ff
   inet 192.168.56.101/28 brd 192.168.56.255 scope global eth1
       valid_lft forever preferred_lft forever
   inet6 fe80::a00:27ff:fe4f:4181/64 scope link
       valid_lft forever preferred_lft forever
```

`ifconfig` (Debian < 8 | Ubuntu < 14.04 )

```
eth0      Link encap:Ethernet  HWaddr 9x:xx:xx:xx:xx:xx
          inet adr:192.168.1.xx  Bcast:192.168.1.255  Masque:255.255.255.0
          adr inet6: fe80::x:xx:xx:xx:xx:xx104/64  Scope:Lien
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:2182 errors:0 dropped:0 overruns:0 frame:0
          TX packets:2144 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 lg file transmission:1000
          RX bytes:1323295 (1.2 MiB)  TX bytes:426001 (416.0 KiB)
          Interruption:252 Adresse de base:0xa000

lo        Link encap:Boucle locale
          inet adr:127.0.0.1  Masque:255.0.0.0
          adr inet6: ::1/128  Scope:Hôte
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:70 errors:0 dropped:0 overruns:0 frame:0
          TX packets:70 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 lg file transmission:0
          RX bytes:5628 (5.4 KiB)  TX bytes:5628 (5.4 KiB)
```

## Serveurs DNS

sudo nano /etc/resolv.conf (voir <https://debian-facile.org/doc/systeme/resolv.conf>)

## Wi-Fi

### Renvoyer le nom du point d'accès wifi

```
sudo iwgetid wlp3s0 --raw
```

### Identification et puissance du signal des points d'accès wifi

```
nmcli dev wifi
```

Pour des informations continuellement mises à jour, on peut utiliser wavemon

```
sudo apt install wavemon
wavemon
```

### Trouver le protocole de sécurité et le type de chiffrement d'un point d'accès

ip link voir les interfaces réseau (repérer l'interface wifi)

```
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN mode DEFAULT group default qlen 1000
   link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP mode DEFAULT group default qlen 1000
   link/ether 50:7b:9d:4d:61:c4 brd ff:ff:ff:ff:ff:ff
3: wlan0: <BROADCAST,MULTICAST> mtu 1500 qdisc noqueue state DOWN mode DEFAULT group default qlen 1000
   link/ether dc:53:60:fd:2b:db brd ff:ff:ff:ff:ff:ff
```

sudo iwlist wlp3s0 scan donne les infos sur points d'accès, protocoles de sécurité (WPA, WPA2, etc.) et modes de chiffrement (TKIP, AES)

### cliquer pour afficher la réponse à cette commande

```
wlan0 Scan completed :
  Cell 01 - Address: 08:87:C6:22:1D:78
    Channel:11
    Frequency:2.462 GHz (Channel 11)
    Quality=66/70 Signal level=-44 dBm
    Encryption key:on
    ESSID:"Proxi-29"
    Bit Rates:1 Mb/s; 2 Mb/s; 5.5 Mb/s; 11 Mb/s; 6 Mb/s
              9 Mb/s; 12 Mb/s; 18 Mb/s
    Bit Rates:24 Mb/s; 36 Mb/s; 48 Mb/s; 54 Mb/s
    Mode:Master
    Extra:tsf=0000046728a98587
    Extra: Last beacon: 244ms ago
    ...
    IE: IEEE 802.11i/WPA2 Version 1
        Group Cipher : TKIP
        Pairwise Ciphers (2) : CCMP TKIP
        Authentication Suites (1) : PSK
    IE: WPA Version 1
        Group Cipher : TKIP
        Pairwise Ciphers (2) : CCMP TKIP
        Authentication Suites (1) : PSK
  Cell 02 - Address: 2A:3A:4D:52:0E:70
    Channel:6
    Frequency:2.437 GHz (Channel 6)
    Quality=70/70 Signal level=-39 dBm
    Encryption key:on
    ESSID:"DIRECT-70-HP M118 LaserJet"
    Bit Rates:6 Mb/s; 9 Mb/s; 12 Mb/s; 18 Mb/s; 24 Mb/s
              36 Mb/s; 48 Mb/s; 54 Mb/s
    Mode:Master
    Extra:tsf=000000561c448ded
    Extra: Last beacon: 684ms ago
    ...
    IE: IEEE 802.11i/WPA2 Version 1
        Group Cipher : CCMP
        Pairwise Ciphers (1) : CCMP
        Authentication Suites (1) : PSK
    ...
```

```

Cell 03 - Address: 44:CE:7D:BF:1B:A4
Channel:1
Frequency:2.412 GHz (Channel 1)
Quality=26/70 Signal level=-84 dBm
Encryption key:on
ESSID:"SFR_1BA0"
Bit Rates:1 Mb/s; 2 Mb/s; 5.5 Mb/s; 11 Mb/s; 18 Mb/s
          24 Mb/s; 36 Mb/s; 54 Mb/s
Bit Rates:6 Mb/s; 9 Mb/s; 12 Mb/s; 48 Mb/s
Mode:Master
Extra:tsf=00000787c58fbc64
Extra: Last beacon: 1116ms ago
...
IE: IEEE 802.11i/WPA2 Version 1
   Group Cipher : CCMP
   Pairwise Ciphers (1) : CCMP
   Authentication Suites (1) : PSK
...

```

## Identifier les machines sur un réseau local

```

# 20240917 testé sur Debian12 / kirin
sudo apt install nmap
sudo nmap -sn 192.168.1.1/24

```

Sans nmap, on peut utiliser `ip neigh` ou ce script python ([source](#)) mais ça ne renvoie pas autant de résultats que nmap

Avec **arp** «Address Resolution Protocol (ARP) is a fundamental part of IP networking. ARP's primary function is to convert 32-bit IP addresses to 48-bit Ethernet MAC addresses. This conversion is crucial because while IP networking requires IP addresses, the underlying Ethernet hardware can only understand MAC addresses.»

```
sudo arp -a #
```

### Avec **arp-scan**

```
sudo apt install arp-scan
arp-scan -l
```

### Avec **ip**

```
ip neigh show
```

```

#!/usr/bin/python3

"""List all hosts with their IP address of the current network."""

import os

out = os.popen('ip neigh').read().splitlines()
for i, line in enumerate(out, start=1):
    ip = line.split(' ')[0]
    h = os.popen('host {}'.format(ip)).read()
    hostname = h.split(' ')[-1]
    print("{:>3}: {} ({}).format(i, hostname.strip(), ip)

```

## Résolution de problèmes

### AppImage et sandbox

Un message d'erreur assez commun lorsqu'on essaie de démarrer un fichier AppImage, créé avec electron : The SUID sandbox helper binary was found, but is not configured correctly. Rather than run without sandboxing I'm aborting now. You need to make sure that `/tmp/.../chrome-sandbox` is owned by root and has mode 4755.

On peut le résoudre en démarrant l'application .AppImage avec l'option `'-no-sandbox'`, exemple :

```

chmod +x ./gb-studio-linux.AppImage
./gb-studio-linux.AppImage --no-sandbox

```

Il existe d'autres possibilités pour résoudre ce problème : <https://stackoverflow.com/a/63788999>

### Réparer une clé USB en lecture seule

```

findmnt # chercher le nom du volume et son type de formatage, ici VFAT
sudo umount /dev/sdb1 # démonter le volume avant de travailler dessus

```

```
sudo dosfsck -a /dev/sdb1 # analyse et réparation d'un volume VFAT
```

Article extrait de : <https://lesporteslogiques.net/wiki/> - **WIKI Les Portes Logiques**

Adresse : [https://lesporteslogiques.net/wiki/ressourcesysteme/commandes\\_linux/start](https://lesporteslogiques.net/wiki/ressourcesysteme/commandes_linux/start)

Article mis à jour: **2025/09/19 11:12**