

Les commandes de base

sudo raspi-config

sudo reboot

Se connecter au Wifi avec le shell :

sudo nano /etc/network/interfaces

Ajouter la ligne :

auto wlan0

Tout en haut du fichier.

Et en bas du fichier, on ajoute les lignes suivantes :

allow-hotplug wlan0

iface wlan0 inet dhcp

wpa-conf /etc/wpa_supplicant/

wpa_supplicant.conf

iface default inet dhcp

Sauvegarder le fichier avec ctrl x puis taper y et appuyer sur entrée.

Ouvrir le fichier de configuration wpa :

nano sudo /etc/wpa_supplicant/wpa_supplicant.conf

Et ajouter les lignes suivantes à la fin du fichier :

network={

ssid="YOUR_NETWORK_NAME" à remplacer par le nom de votre réseau wifi

psk="YOUR_NETWORK_PASSWORD" à remplacer par votre mot de passe

proto=RSN

key_mgmt=WPA-PSK

pairwise=CCMP

auth_alg=OPEN

}

Puis sauvegarder le fichier : ctrl x, y, entrée

Et enfin rebooter votre pi : sudo reboot

Pour tester la connection : ping www.google.com

Ctrl c pour arrêter le ping

Installer SAMBA :

sudo apt-get install samba samba-common-bin

(Si ça ne marche pas : sudo apt-get update && sudo apt-get upgrade

Ou apt-get install smb)

Puis appuyer sur y et entrée

- Sécuriser la connection au NAS avec un log in et mot de passe

sudo nano /etc/samba/smb.conf

Descendez jusqu'à la section authentication

Juste en dessous du titre de la section, on ajoute la ligne

security = user

Sauvegarder le fichier config avec ctrl x, y et entrée

Puis redémarrer Samba pour mettre à jour la configuration

sudo /etc/init.d/samba restart (~~sudo /etc/init.d/nmbd restart~~ , sudo /etc/init.d/smbd restart)

On va configurer des espaces de stockages privés

Réouvrir le fichier de configuration Samba

```
sudo nano /etc/samba/smb.conf
```

On cherche la section Homes, dans laquelle vous devriez trouver read only = yes et on change yes en no pour nous permettre d'effectuer l'action souhaitée sur les fichiers, pas seulement les ouvrir et les lire.

On sauvegarde le fichier par ctrl x, y et entrée.

On redémarre Samba :

```
sudo /etc/init.d/samba restart
```

Par défaut, c'est l'utilisateur pi qui est défini, on va maintenant l'ajouter aux utilisateurs de Samba avec la commande :

```
sudo smbpasswd -a pi
```

Puis on vous demande un mot de passe ainsi une confirmation, choisissez le mot de passe que vous voulez (**raspberry** par exemple), et appuyez sur entrée.

Ajouter de nouveaux utilisateurs :

```
sudo useradd john -m -G users
```

Entrer ensuite la commande :

```
sudo passwd john
```

On vous demandera de rentrer deux fois un mot de passe pour l'utilisateur john

Maintenant, il faut autoriser cet utilisateur à être un utilisateur de Samba :

```
sudo smbpasswd -a john
```

On vous demande de rentrer deux fois un mot de passe (celui créé pour john)

Création d'un espace public sur le NAS :

Créer ce répertoire avec les commandes suivantes

```
sudo mkdir /home/shares
```

```
sudo mkdir /home/shares/public
```

```
sudo chown -R root:users /home/shares/public
```

```
sudo chmod -R ug=rwx, o=rx /home/shares/public
```

On va aller éditer le fichier de config Samba :

```
sudo nano /etc/samba/smb.conf
```

Tout en bas du fichier, on va ajouter une section relative à ce dossier public. lignes à écrire sont :
[public]

```
comment = Public Storage
```

```
path = /home/shares/public
```

```
valid users = @users
```

```
force group = users
```

```
create mask = 0660
```

```
directory mask = 0771
```

```
read only = no
```

Sauvegarder la config avec ctrl x, y et entrée

Redémarrer Samba :

```
sudo /etc/init.d/samba restart
```

On va ajouter plus d'espace de stockage sur le NAS (utiliser u disc dur ou clé USB) :

On doit ajouter les disc dur 1 à 1.

Brancher un disc dur sur la raspberry

Lancer la commande :

```
dmesg
```

Tout en bas, notre disc apparait et un nom lui a été assigné (par exp sda1) ...(1)

```
sudo mkdir /home/shares/public/disk1
```

(disk1 à remplacer par le nom du disc que vous souhaitez assigné à ce disc, stockage_SN)

`sudo mount /dev/sda1 /home/shares/public/disk1` à remplacer sda1 par le nom assigné en (1)

Si nécessaire, à refaire pour chaque disc de stockage.

Inconvénient : A chaque redémarrage de la carte, il faut remonter le disc avec les commandes précédentes.

Solution :

Ouvrir le fichier fstab

`sudo nano /etc/fstab` (à faire très attention aux fautes : risque de bloquer la raspberry)

Une fois dans le fichier, ajouter la ligne suivante à la fin du fichier en laissant une ligne vide avant.

`/dev/sda1 /home/shares/public/disk1 auto noatime 0 0`

Cette ligne est relative à un seul disc dur, il faut ajouter à chaque fois cette ligne pour chaque disc dur.

Sauvegarder : ctrl x, y, entrée

Se connecter au NAS :

a. Aller dans « Ce PC » (Ordinateur), cliquer sur connecter un lecteur réseau, remplir le champ « Dossier » par `\\RASPBERRYPI\public` pour accéder au dossier public, on nous demande un login (john) et un mot de passe (raspberry). On peut stocker des fichiers dans disk1 (disc dur) ou en dehors (carte sd du raspberry).

b. Aller dans « Ce PC » (Ordinateur), cliquer sur connecter un lecteur réseau, remplir le champ « Dossier » par `\\RASPBERRYPI\john` pour accéder au dossier privé de john, on nous demande un login (john) et un mot de passe (celui créé pour john).

Remarque : en cas de difficultés à se connecter au raspberry, récupérer l'adresse IP de raspberry en utilisant ifconfig

Problème d'accès à `\\RASPBERRYPI\public\disk1`

rebooter le Raspberry et relancé les commandes :

`"sudo chown -R root:users /home/shares/public"`

`"sudo chmod -R ug=rwx,o=rx /home/shares/public".`

En cas de problème de droit pour accéder au NAS depuis Windows ou autre, il faut installer le paquet ntfs-3g (commande: `sudo apt install ntfs-3g - y`) puis un reboot de la Pi pour que ça fonctionne directement. Ou formater le disque en FAT32.

Pour utiliser plusieurs disques sans problème, à savoir les disques potentiellement rebranchés à des endroits différents, il existe une méthode de montage par identifiant unique (un peu équivalent à l'adresse mac en réseau).

Dans la console, on retrouve tous les périphériques branchés avec la commande : `blkid`

puis avec un peu de logique, on retrouve quel identifiant correspond à quel périphérique

On modifie le fichier de config comme précédemment avec

`sudo /etc/fstab`

et la ligne donnée en exemple

`/dev/sda1 /home/shares/public/disk1 auto noatime 0 0`

devient

`/dev/disk/by-uuid/<identifiant trouvé> /home/shares/public/disk1 auto noatime 0 0`

(Il ne faut pas mettre les guillemets)

Si en montant le disque dur externe, s'il indique une erreur ntfs supérieur à 2To; installer un paquet de plus avant de monter le disc.

`sudo apt-get install ntfs-3g`

Problème d'accès au disc dur : `sudo mkfs.ext4 /dev/sda1`

créer un dossier sur sda1 accessible uniquement depuis John ?

`sudo mkdir /home/shares/John/disk2`

```
sudo mount /dev/sda1 /home/shares/John/disk2
dans fstab : /dev/sda1 /home/shares/John/disk2 auto noatime 0 0
```

La syntaxe d'une commande est de la forme suivante : **COMMANDE** **OPTIONS** **ARGUMENTS**

La commande étant la commande elle-même, suivie d'une ou plusieurs options **facultatives** qui servent à modifier le comportement de la commande tandis que les arguments, **facultatifs** aussi, représentent les objets sur lesquels la commande agit.

Un exemple concret peut être : **ls -a /home** ou **ls** est la commande qui permet de lister l'ensemble des fichiers d'un répertoire, en l'occurrence, le répertoire **/home** et l'option **-a** permet d'inclure les fichiers cachés dans le résultat.

Voici une liste des commandes les plus populaires et que vous devriez connaître lors de la manipulation d'un système Linux :

1. La Commande **pwd**

Cette commande permet d'afficher l'emplacement où on se situe actuellement dans la hiérarchie FHS.

```
[root@centos ~]# pwd
/root
```

2. La Commande **cd**

Cette commande permet de changer de répertoire courant et de se situer sur un autre

```
[root@centos ~]# cd /home/
[root@centos home]# pwd
/home
```

Pour naviguer dans les fichiers et répertoires de Linux, utilisez la commande **cd**. Elle nécessite soit le chemin d'accès complet, soit le nom du répertoire, selon le répertoire de travail dans lequel vous vous trouvez.

Disons que vous êtes dans **/home/utilisateur/Documents** et que vous voulez aller dans **Photos**, un sous-répertoire de **Documents**. Pour ce faire, il vous suffit de taper la commande suivante : **cd Photos**.

Un autre scénario est possible si vous voulez passer à un répertoire complètement nouveau, par exemple, **/home/utilisateur/Films**. Dans ce cas, vous devez taper **cd** suivi du chemin absolu du répertoire : **cd /home/utilisateur/Films**.

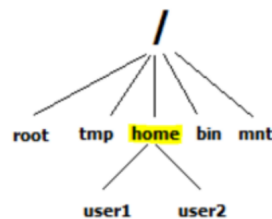
Il existe quelques raccourcis pour vous aider à naviguer rapidement :

- **cd ..** (avec deux points) pour se déplacer d'un répertoire vers le haut
- **cd** pour aller directement au dossier principal (home)
- **cd-** (avec un tiret) pour passer à votre répertoire précédent

Par ailleurs, le shell de Linux est sensible à la casse. Vous devez donc taper les noms des répertoires exactement comme ils sont.

3. La Commande **ls**

Permet de lister les fichiers disponibles dans un répertoire, si appelé sans arguments, **ls** liste les fichiers du répertoire courant.



```
[root@centos home]# ls
user1 user2
```

=> Ainsi, dans le répertoire home, deux fichiers existent et sont : user1 et user2

Il existe des variantes que vous pouvez utiliser avec la commande **ls** :

- **ls -R** énumérera également tous les fichiers dans les sous-répertoires
- **ls -a** affichera les fichiers cachés
- **ls -al** listera les fichiers et les répertoires avec des informations détaillées comme les autorisations, la taille, le propriétaire, etc.

4. La Commande **mkdir**

Cette commande permet de créer un répertoire.

```
[root@centos home]# cd /tmp
[root@centos tmp]# pwd
/tmp
[root@centos tmp]# mkdir repertoiredetest
[root@centos tmp]# ls
repertoiredetest
fichiertemp
```

=> Dans cet exemple, on s'est déplacé sous le répertoire **tmp** disponible sous la racine /, affiché notre emplacement grâce à la commande **pwd**, puis créé un répertoire (dossier) nommé **repertoiredetest** et le visualisé avec la commande **ls**.

Il existe également des commandes **mkdir** supplémentaires :

- Pour générer un nouveau répertoire à l'intérieur d'un autre répertoire, utilisez cette commande de base de Linux **mkdir Music/Nouveau**

- utiliser l'option **p** (parents) pour créer un répertoire entre deux répertoires existants. Par exemple, **mkdir -p Musique/2020/Nouveau** créera le nouveau répertoire « **2020** ».

5. La Commande **rmdir**

Cette commande permet de supprimer un répertoire.

```
[root@centos tmp]# rmdir repertoiredelest  
[root@centos tmp]# ls  
fichier1  
fichier2
```

=> Avec cette commande, on vient de supprimer le répertoire qu'on vient de créer et la commande **ls** nous l'a confirmé.

6. La Commande **touch**

Cette commande permet de changer la date du dernier accès ou modification d'un fichier, mais permet également de créer un fichier vide.

```
[root@centos tmp]# touch fichier  
[root@centos tmp]# ls  
fichier  
fichier1  
fichier2
```

=> De même, mais cette fois, on a créé un fichier nommé *fichier*.

7. La Commande **cp**

Cette commande permet de copier un fichier ou un répertoire.

```
[root@centos tmp]# cp fichier fichier2  
[root@centos tmp]# ls  
fichier  
fichier2  
fichier1  
fichier3
```

=> Avec la commande **cp**, on a copié le fichier *fichier* en le nommant *fichier2*.

8. La Commande **rm**

Cette commande permet de supprimer un fichier ou un répertoire. Cette commande est à utiliser avec précaution car avec l'option **-f** ou **-rf**, cette commande peut endommager voire supprimer tout votre système de fichiers de manière irréversible.

```
[root@centos tmp]# rm -f fichier fichier2  
[root@centos tmp]# ls  
fichier1  
fichier3
```

=> La commande **rm** nous a permis dans cet exemple de supprimer les deux fichiers précédemment créés.

9 La Commande **mv**

Cette commande sert à renommer ou déplacer un fichier ou un répertoire.

Dans l'exemple suivant, on va créer un fichier nommé "fichier3", et à l'aide de la commande **mv**, on va le renommer en "fichier4" et le déplacer dans le répertoire "/home" en une seule fois.

```
[root@centos tmp]# touch fichier3
[root@centos tmp]# mv fichier3 /home/fichier4
[root@centos tmp]# ls /home/
fichier4 user1 user2
```

11. La Commande **cat**

Cette commande permet d'afficher le contenu d'un fichier.

```
[root@ip8 tmp]# cat /home/fichier4
[root@ip8 tmp]#
```

le fichier "fichier4" est vide, donc la commande "cat" ne renverra aucun résultat, dans un autre cas on pourrait avoir :

```
[root@centos tmp]# cat /etc/resolv.conf
# Generated by NetworkManager
nameserver 8.8.8.8
```

Souvenez-vous que **/etc/** contient principalement des fichiers de configuration, ici, le fichier **resolv.conf** est le fichier qui détient les informations **DNS** de notre système (8.8.8.8 est l'adresse IP du serveur DNS de google).

12. La Commande **echo**

Cette commande permet d'afficher une ligne.

```
[root@centos tmp]# echo "tutoriel Linux de sitedetout"
tutoriel Linux de sitedetout
```

=> **echo** permet aussi d'écrire du contenu dans un fichier moyennant le signe ">" pour écraser le contenu du fichier ou ">>" pour suffixer le contenu du fichier.

```
[root@centos tmp]# cat /home/fichier4
[root@centos tmp]# echo contenu > /home/fichier4
[root@centos tmp]# cat /home/fichier4
contenu
[root@centos tmp]# echo contenu2 > /home/fichier4
[root@centos tmp]# cat /home/fichier4
```

```
contenu2
[root@centos tmp]# echo contenu3 >> /home/fichier4
[root@centos tmp]# cat /home/fichier4
contenu2
contenu3
```

13. La Commande **man**

Cette commande permet d'afficher le manuel d'aide de n'importe quelle autre commande disponible, ex : **man ls**

Tapez la lettre **q** pour fermer la page du manuel.

14. La Commande **which**

Cette commande permet de visualiser l'emplacement d'une commande en effectuant une recherche dans différents répertoires.

```
[root@centos tmp]# which pwd

/bin/pwd

[root@centos tmp]# which mkdir

/bin/mkdir
```

15. La Commande **more**

La commande **more** permet de visualiser le contenu d'un fichier page à page. Utilisez la touche entrée ou espace pour avancer l'affichage du contenu de votre fichier.

```
[root@centos tmp]# touch fichier

[root@centos tmp]# echo "contenu" >> fichier

[root@centos tmp]# more fichier

contenu
```

16. La Commande **head**

La commande **head** permet d'afficher le début d'un fichier (par défaut, les 10 premières lignes). Pour démontrer son fonctionnement, j'ai créé un fichier nommé fichier2 avec un contenu de 20 lignes.

```
[root@centos tmp]# cat fichier2
ligne 1
ligne 2
ligne 3
ligne 4
ligne 5
ligne 6
```


...

...

ligne 17

ligne 18

ligne 19

ligne 20

```
[root@centos tmp]# head fichier2
```

ligne 1

ligne 2

ligne 3

ligne 4

ligne 5

ligne 6

ligne 7

ligne 8

ligne 9

ligne 10

17. La Commande **tail**

La commande **tail** permet d'afficher la fin d'un fichier (par défaut, les 10 dernière lignes)

```
[root@centos tmp]# tail fichier2
```

ligne 11

ligne 12

ligne 13

ligne 14

ligne 15

ligne 16

ligne 17

ligne 18

ligne 19

ligne 20

18. Commande **sudo**

Abréviation de « **SuperUser Do** », cette commande vous permet d'effectuer des tâches qui nécessitent des autorisations administratives ou de root. Cependant, il n'est pas conseillé d'utiliser cette commande pour un usage quotidien car une erreur pourrait facilement se produire si vous avez fait quelque chose de incorrect.

19. Commande **df**

Utilisez la commande **df** pour obtenir un rapport sur l'utilisation de l'espace disque du système, indiquée en pourcentage et en Ko. Si vous voulez voir le rapport en mégaoctets, tapez **df -m**.

20. Commande **chmod**

Chmod est une autre commande Linux, utilisée pour modifier les permissions de lecture, d'écriture et d'exécution des fichiers et des répertoires. Comme cette commande est assez compliquée, vous pouvez lire le [tutoriel complet](#) afin de l'exécuter correctement.

21. Commande chown

Sous Linux, tous les fichiers sont la propriété d'un utilisateur spécifique. La commande **chown** vous permet de changer ou de transférer la propriété d'un fichier à un utilisateur spécifique. Par exemple, **chown linuxuser2 fichier.ext** fera de **linuxuser2** le propriétaire du **fichier.ext**.

22. Commande ping

Utilisez la commande **ping** pour vérifier votre état de connectivité à un serveur. Par exemple, en entrant simplement **ping google.com**, la commande vérifiera si vous êtes en mesure de vous connecter à Google et mesurera également le temps de réponse.

23. Commande wget

La ligne de commande Linux est super utile – vous pouvez même télécharger des fichiers sur Internet à l'aide de la commande **wget**. Pour ce faire, il suffit de taper **wget** suivi du lien de téléchargement.

24. Commande uname

La commande **uname**, abréviation de Unix Name, imprimera des informations détaillées sur votre système Linux comme le nom de la machine, le système d'exploitation, le noyau, etc.

25. Commande top

Comme un terminal équivalent au gestionnaire de tâches dans Windows, la commande **top** affichera une liste des processus qui sont en cours d'exécution et la quantité de CPU utilisée par chaque processus. Il est très utile de surveiller l'utilisation des ressources du système, en particulier de savoir quel processus doit être arrêté en cas de surconsommation de ressources.

26. Commande history

Lorsque vous utilisez Linux depuis un certain temps, vous remarquerez rapidement que vous pouvez exécuter des centaines de commandes chaque jour. Ainsi, l'exécution de la commande **history** est particulièrement utile si vous voulez revoir les commandes que vous avez entrées auparavant.

27. Commande man

Confus quant à la fonction de certaines commandes Linux ? Ne vous inquiétez pas, vous pouvez facilement apprendre à les utiliser directement depuis le shell de Linux en utilisant la commande **man**. Par exemple, en entrant la commande **man tail**, vous verrez les instructions manuelles de la commande tail.

28. Commande zip, unzip

Utilisez la commande **zip** pour compresser vos fichiers dans une archive zip, et utilisez la commande **unzip** pour extraire les fichiers zippés d'une archive zip.

29. Commande **hostname**

Si vous voulez connaître le nom de votre hôte/réseau, il vous suffit de taper **hostname**. En ajoutant un **-I** à la fin, vous obtiendrez l'adresse IP de votre réseau.

30. Commande **useradd**, **userdel**

Puisque Linux est un système multi-utilisateurs, cela signifie que plusieurs personnes peuvent interagir avec le même système en même temps. **useradd** est utilisé pour créer un nouvel utilisateur, tandis que **passwd** est l'ajout d'un mot de passe au compte de cet utilisateur. Pour ajouter une nouvelle personne nommée John, tapez **useradd John** et ensuite pour ajouter son mot de passe, tapez **passwd 123456789**.

La suppression d'un utilisateur est très similaire à l'ajout d'un nouvel utilisateur. Pour supprimer le compte d'un utilisateur, tapez, **userdel NomUtilisateur**

Conseils et astuces

Utilisez la commande **clear** pour nettoyer le terminal s'il est encombré par trop de commandes Linux déjà effectuées.

Essayez le bouton **TAB** pour remplir automatiquement ce que vous tapez. Par exemple, si vous avez besoin de taper Documents, commencez à taper une commande (disons **cd Docu**, puis appuyez sur la touche **TAB**) et le terminal remplira le reste, ce qui vous donnera **cd Documents**.

Ctrl+C et **Ctrl+Z** sont utilisés pour arrêter toute commande qui fonctionne actuellement. Ctrl+C arrêtera et terminera la commande, tandis que Ctrl+Z mettra simplement la commande en pause.

Si vous bloquez accidentellement votre terminal en utilisant **Ctrl+S**, il suffit de le débloquent avec **Ctrl+Q**.

Ctrl+A vous amène au début de la ligne tandis que **Ctrl+E** vous amène à la fin.

Vous pouvez exécuter plusieurs commandes en une seule en utilisant « ; » pour les séparer. Par exemple, **Commande1; Commande2; Commande3**. Ou utilisez **&&** si vous souhaitez que la commande suivante ne soit exécutée que lorsque la première a réussi.