# Fondamentaux

Apprends ça et reviens après !

- [Fondamentaux] Système de fichier
- [Fondamentaux] Cron
- [Fondamentaux] Gestion des droits
- [Fondamentaux] Tester la connectivité entre deux machines
- [Fondammentaux] Partitionnement

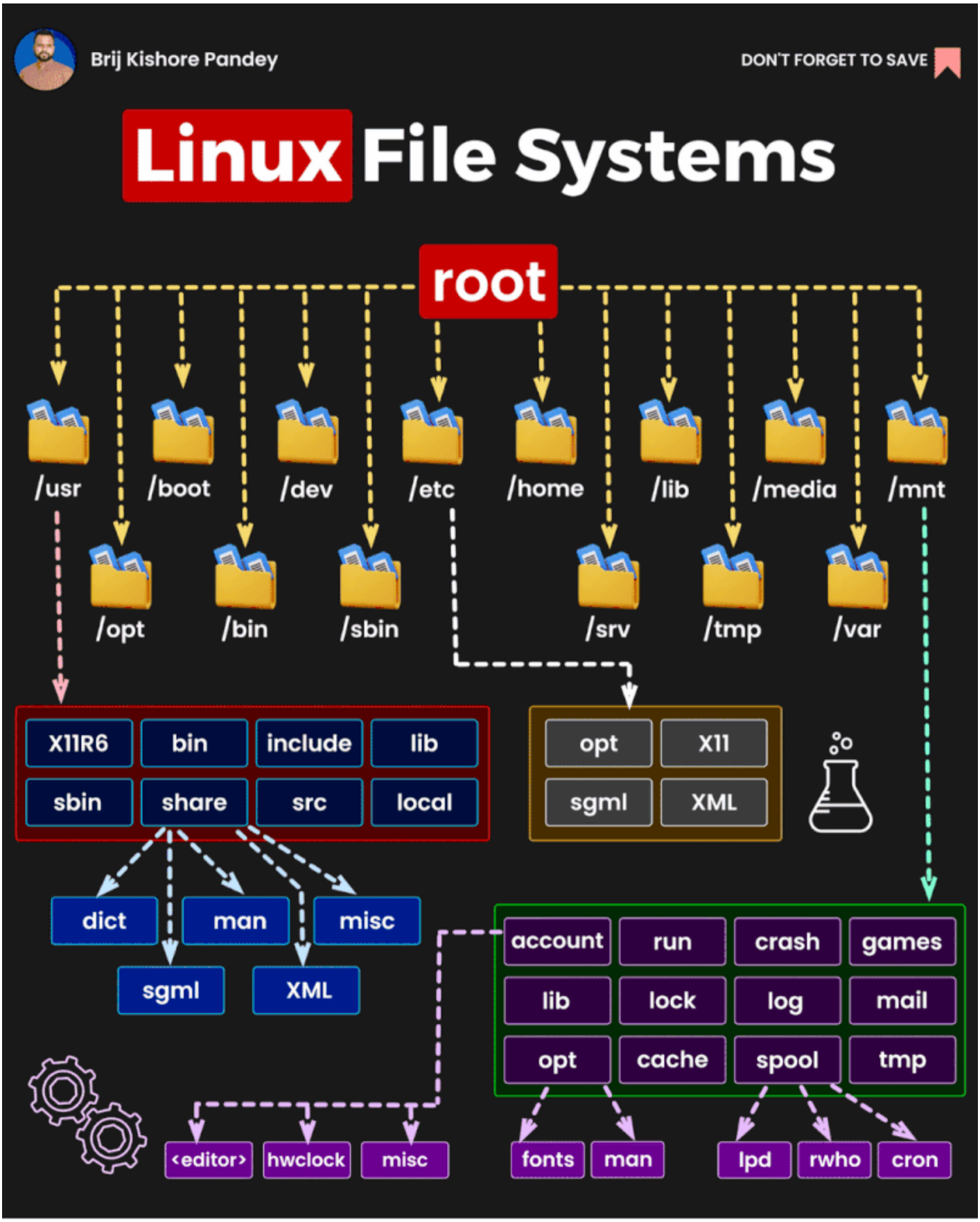
# [Fondamentaux] Système de fichier

## Introduction

Linux comme beaucoup d'autres systèmes repose sur un système de fichier hiérarchique.

On parle aussi d'arborescence.

## Le descriptif




/

| Point de montage | Fonction | Logo | Descriptif |
| --- | --- | --- | --- |

| | | | |
|---|---|---|---|
| /usr | Utilisateur | □□ | Ce répertoire contient des données supplémentaires, des programmes et des bibliothèques utilisés par les utilisateurs et les applications. |
| /boot | Démarrage | □□ | C'est là que se trouvent les fichiers nécessaires au démarrage du système, y compris le noyau Linux. |
| /dev | Dispositifs | □□ | Les fichiers spéciaux représentant les périphériques sont stockés ici. |
| /etc | Configuration | □□ | Les fichiers de configuration système résident dans ce dossier. Vous y trouverez des informations cruciales sur le système et les applications. |
| /home | Domicile | □□ | Les dossiers personnels des utilisateurs sont stockés ici, chacun ayant son propre répertoire avec son nom d'utilisateur. |
| /lib | Bibliothèques | □□ | Les bibliothèques partagées essentielles pour le fonctionnement du système se trouvent ici. |
| /media | Média | □□ | Les périphériques externes tels que les clés USB et les disques optiques sont montés ici. |
| /mnt | Montage | □□ | Ce répertoire est utilisé pour monter temporairement des systèmes de fichiers, tels que des lecteurs réseau. |
| /opt | Optionnel | □□ | Les logiciels tiers, qui ne font pas partie de la distribution par défaut, sont souvent installés ici. |
| /bin | Binaire | □□ | Les exécutables essentiels du système, utilisables par tous les utilisateurs, sont stockés ici. |

| | | | |
|---|---|---|---|
| /sbin | Super Binaire | □□ | C'est l'emplacement des programmes essentiels destinés à l'administrateur système. |
| /srv | Service | □□ | Les données de service, telles que les fichiers de site Web, peuvent être stockées ici. |
| /tmp | Temporaire | □□ | Les fichiers temporaires sont stockés ici, mais ce répertoire est nettoyé au démarrage. |
| /var | Variables | □□ | Les données variables telles que les journaux, les bases de données et les fichiers temporaires résident ici. Il est essentiel pour surveiller l'état du système. |

# [Fondamentaux] Cron

## Introduction

Cron est un **planificateur de tâche** disponible sur les distributions Linux.

Il peut permettre l'exécution de binaire ou de script notamment pour vos sauvegardes ou vos programmes.




## Configuration

Cron se configure depuis le fichier **/etc/crontab** où chaque ligne est une tâche planifiée. Voici un pense bête explicatif :

# Cron reference sheet for sysadmins

## Format

```
Min  Hour  Day  Mon  Weekday

*    *     *    *    *     script/command to be executed

┬    ┬     ┬    ┬    ┬
│    │     │    │    └──── Day of Week      ( 0=Sun . . 6=Sat )
│    │     │    └───────── Month            ( 1 . . 12 )
│    │     └────────────── Day of Month     ( 1 . . 31 )
│    └──────────────────── Hour             ( 0 . . 23 )
└───────────────────────── Minute           ( 0 . . 59 )
```

| | | |
|---|---|---|
| **Minutes** | 0 . . 59 | the command/script would be executed at the specified minute. |
| **Hours** | 0 . . 23 | the command/script would be executed at the specified hour. |
| **Days** | 1 . . 31 | the days of the months in which the script or command would be executed. |
| **Months** | 1 . . 12 | the month in which the script would be executed. |
| **Weekdays** | 0 . . 6 | the days of the week in which the script gets executed. 0 is Sunday. |

## Special strings

| | |
|---|---|
| @reboot | command will be executed once at system startup (non-standard). |
| @hourly | command will be executed once an hour, same as ("0 * * * *") but non-standard. |
| @daily | command will be executed once each day, same as ("0 0 * * *") but non-standard. |
| @midnight | same as @daily but also non-standard. |
| @weekly | command will be executed once every week, same as ("0 0 * * 0") but non-standard. |
| @monthly | command will be executed once every month, same as ("0 0 1 * *") but non-standard. |
| @yearly | command will be executed once every year, same as ("0 0 1 1 *") but non-standard. |

## Examples

| | | |
|---|---|---|
| 0 * * * * | /opt/backup.sh | perfom a system backup every hour. |
| */7 * * * * | /opt/ping.sh | check if the remote server is online every 7 minutes. |
| 0 */6 * * * | /opt/emptytrash.sh | empty trush every 6 hours. |
| 20 14 * * * | /opt/upgrade | upgrade the system at 14:20 PM of every day. |
| 5 9 * 4 * | /opt/upgrade | upgrade the system at 09:05 AM in April. |
| 20 14 * * ? | /opt/update.sh | update system At 14:20 PM of every day. |
| 6 11 * * 3 | /opt/upgrade.sh | upgrade the system at 11:06 AM of every Wednesday. |
| 0 22 * * 1-5 | /opt/upgrade.sh | upgrade the system at 22:00 PM on every day-of-week from Monday through Friday. |
| 0 0 * * 2 | /opt/upgrade.sh | upgrade the system at midnight (00:00) of ever Tuesday. |
| 10 8 * * 4L | /opt/monitor.sh | monitor the system at 08:10 AM on the last Thursday of every month. |
| 15 0 * * 4#2 | /opt/upgrade | upgrade the system at at 00:15 AM on the second Thursday of every month. |
| 0 0 0 1 * * | /opt/backup.sh | perfom a sys backup every 1st of month (monthly). |
| 0 0 0 1 1 * | /opt/backup.sh | perfom a sys backup every 1st of january (yearly). |
| 5 12 * * 6 | /opt/emptytrash.sh | clears the trush at 12:05 PM on Sunday. |
| @reboot | /opt/backup.sh | perfom a system backup at reboot. |

## Crontab

| | |
|---|---|
| crontab -e | Edit or create a crontab file if doesn't already exist. |
| crontab -l | Display the crontab file. |
| crontab -r | Remove the crontab file. |
| crontab -u username -l | Display another user's crontab file. |
| crontab -u username -e | Edit another user's crontab file. |
| crontab -v | Display the last time you edited your crontab file. |



## Special characters

| | |
|---|---|
| Asterik (*) | this operator is used to represent all potential values in a field. Write an asterisk "*" in the Minute column, for example, if you want your cron job to execute every minute. |
| Hyphen (-) | to determine a range of values, use this operator. For example, if you want to set up a cron job from Monday through Friday, simply write 1-5 in the weekday column. |
| Slash (/) | to split a value, use this operator. For instance, if you want a script to run every 6 hours, enter */6 in the Hour field. |
| Comma (,) | to list numerous values, use this operator. Writing 1,5 in the Day of the week field, for example, will schedule the task to be executed every Monday and Friday. |
| Last (L) | this operator can be used in the month and weekday fields. Writing 6L in the day-of-week field, for example, signifies the last Saturday of the month. |
| Weekday (W) | this operator is used to get the closest weekday from a given time. If the 1st of the month is a Saturday, for example, entering 1W in the day-of-month field will execute the command on the following Monday (the 3rd). |
| Hash (#) | it is only permitted for the Day Of Week field, which must be followed by a number between 1 and 5. For instance, 5#2 denotes "the second Friday" of a given month. |
| Question mark (?) | can be used instead of '*' in the Day of Month and Day of Week fields. Use this operator to enter "no specified value" for the "day of the month" and "day of the week" fields. |

sysxplore.com

# [Fondamentaux] Gestion des droits

# Introduction

Linux permet une gestion fine des droits sur les fichiers et les dossiers du système. Cette page ne traitera pas de sudo qui a déjà une page dédiée.

# Commandes

## ls -l

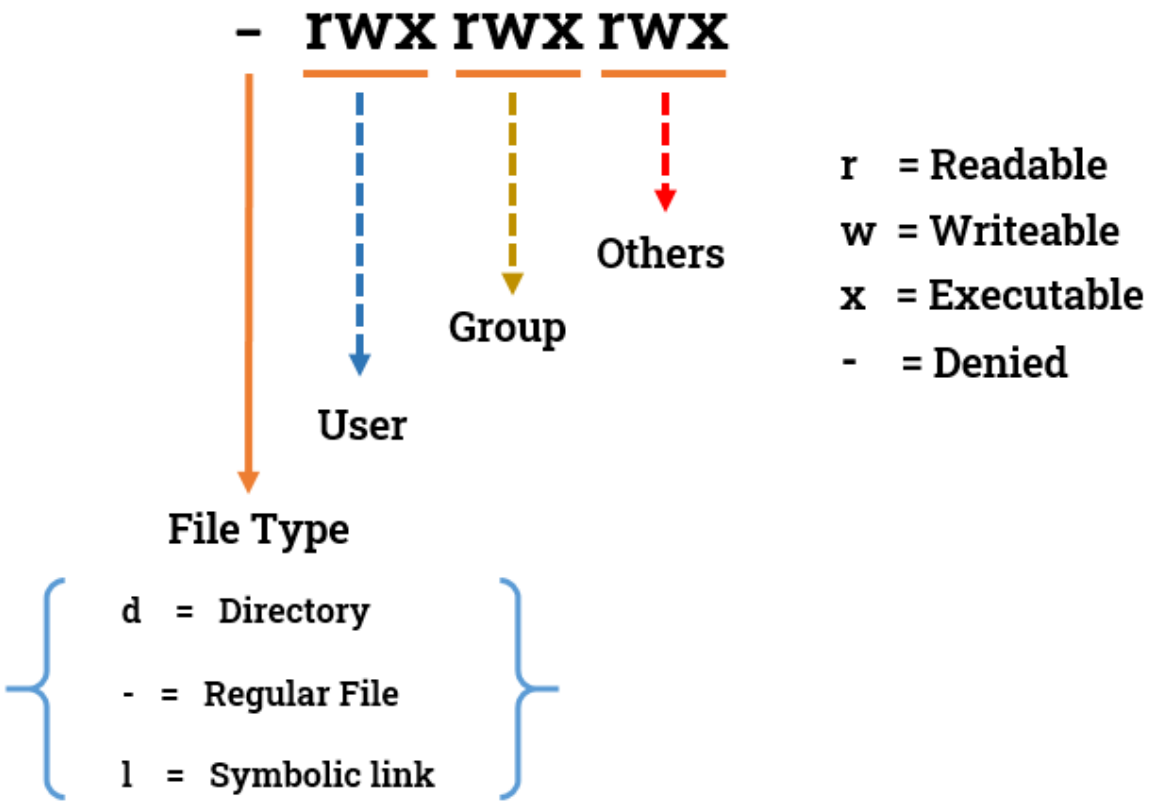
Permet d'afficher le ou les fichiers avec leurs droits associés :

```
ls -l
total 38588
-rw-r--r-- 1 elieroc elieroc   165711  7 oct.  21:32  acl.jpeg
-rw-r--r-- 1 elieroc elieroc   133769  3 oct.  20:44  AGEFIPH.pdf
-rw-r--r-- 1 elieroc elieroc   339479 25 sept. 23:03 'Block devices.pdf'
-rw-r--r-- 1 elieroc elieroc 29345747 27 sept. 20:12 'CCNA Trimestre 1 Partie 1.pdf'
-rw-r--r-- 1 elieroc elieroc   296675  7 oct.  21:32  cron.jpeg
-rw-r--r-- 1 elieroc elieroc   546179 26 sept. 21:50 'DNS Bind.pdf'
-rw-r--r-- 1 elieroc elieroc  1630995  7 oct.  21:31  filesystem.gif
-rw-r--r-- 1 elieroc elieroc   126952  7 oct.  21:39  gnu.png
-rw-r--r-- 1 elieroc elieroc    32858  3 oct.  20:40 'ITRUST - Analyste Threat Intelligence
H_F.docx'
-rw-r--r-- 1 elieroc elieroc  1597123  7 oct.  21:31  kerberos.gif
-rw-r--r-- 1 elieroc elieroc   319746 25 sept. 22:20 'Le pain quotidien.pdf'
-rw-r--r-- 1 elieroc elieroc   144785 27 sept. 22:24  marteau.jpg
-rw-r--r-- 1 elieroc elieroc  4428842 25 sept. 21:44  RAID.pdf
-rw-r--r-- 1 elieroc elieroc   259856 25 sept. 21:51 'Système de fichier.pdf'
drwxr-xr-x 1 elieroc elieroc     1054 25 sept. 22:22 'TP remise à niveau + corrigés'
```

```
-rw-r--r-- 1 elieroc elieroc    93176 25 sept. 22:22 'TP remise à niveau + corrigés.zip'
-rw-r--r-- 1 elieroc elieroc    17688  7 oct.  21:40  tux.webp
```

La première colonne indique les droits et le type de fichier, la troisième indique l'utilisateur propriétaire et la quatrième le groupe propriétaire.

Voici le fonctionnement de la première colonne :




## Chown

Permet de définir le propriétaire du fichier.

```
chown [-R] <USER>:<GROUP> <FILE|DIRECTORY>
```

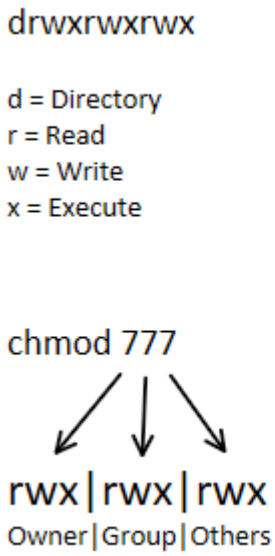
Remarque : L'option **-R** permet la récursivité et ainsi d'appliquer l'opération à tous les fichiers et dossiers contenus dans le dossier indiqué.

## Chmod

Permet de définir les droits du fichier.

```
chmod [-R] <RIGHTS> <FILE>
```

Le champ RIGHTS doit être remplacé par un motif spécifique désignant les droits que vous souhaitez affecter :




| | | |
|---|---|---|
| 7 | rwx | 111 |
| 6 | rw- | 110 |
| 5 | r-x | 101 |
| 4 | r-- | 100 |
| 3 | -wx | 011 |
| 2 | -w- | 010 |
| 1 | --x | 001 |
| 0 | --- | 000 |

Voici un exemple :

```
chmod -R w-- myDirectory/
```

Ici, on autorise seulement le propriétaire à pouvoir modifier tous les fichiers et dossiers contenus dans le dossier **myDirectory** (et le dossier lui-même).

# ACL

Ils permettent de définir des droits spécifiques à des utilisateurs ou des groupes d'utilisateurs :

# ACL 101

1. ACL is a superset of the regular permissions.

First, create a directory...

```
$ mkdir dir
```

Regular permissions:

```
$ ls -ld dir
drxw------
```

ACL permissions:

```
$ getfacl dir
USER:    me     rwx
GROUP:   me     ---
OTHER:          ---
```

Both permissions are the same, let's change that...

2. ACL allows adding permission for specific users and groups

```
$ setfacl -m u:dude:r dir
```

Regular permissions:

```
$ ls -ld dir
drxwr------+
```

ACL exists

This is now the "mask", aka the union of:
- GROUP,
- any custom ACL user,
- any custom ACL group.

ACL permissions:

```
$ getfacl dir
USER:    me     rwx
user:    dude   r--
GROUP:   me     ---
mask:           r--
OTHER:          ---
```

- The "mask" is the mechanism that keeps "regular permissions" and ACL in sync.
- It's updated automatically when running 'chmod' or 'setfacl'

# [Fondamentaux] Tester la connectivité entre deux machines

## Introduction

À des fins de diagnostic réseau, il peut être utile de tester un flux.

Pour cela, vous pouvez utiliser le protocole ICMP ou netcat et nmap pour des flux TCP/UDP.

## TCPdump

Commencez par installer le paquet **tcpdump** :

```
sudo apt install -y tcpdump
```

Pour l'exemple, nous allons tester la connectivité entre une prmeière machine qui a pour IP **192.168.192.1** et une deuxième machine qui a pour IP **192.168.192.129**.

On va donc écouter les requêtes ICMP grâce à tcpdump avec la commande suivante sur la deuxième machine :

```
tcpdump icmp -i <INTERFACE>
```

Puis on peut lancer un ping sur la première machine :

```
ping 192.168.192.129
```

## Netcat

Installez **netcat** :

```
sudo apt install -y netcat
```

## TCP

Pour tester un flux TCP sur un port particulier vous pouvez utiliser la commande suivant :

```
nc -zv <IP> <PORT>
```

## UDP

Pour tester un flux UDP sur un port particulier vous pouvez utiliser la commande suivant :

```
nc -zv -u <IP> <PORT>
```

# Nmap

Installez **nmap** :

```
sudo apt install -y nmap
```

## TCP

```
nmap -p <PORT> <IP>
```

## UDP

```
nmap -sU -p <PORT> <IP>
```

# [Fondammentaux] Partitionnement

## Introduction

Le partitionnement sous Linux est essentiel et permet de "découper" votre disque.

Ici pour l'exemple nous allons créer une simple partition primaire au format ext4 sur le disque /dev/sda.



## Manuel

# Création de la partition

On peut faire avec **parted** :

```
parted /dev/sda mklabel gpt mkpart <LABEL> ext4 1MiB 100%
```

> Remplacez **<LABEL>** par le nom que vous souhaitez donner à votre partition.

On peut aussi faire avec **fdisk** :

```
fdisk /dev/sda
```

Puis exécutez cette enchaînement de touche :

**N / P / ENTER / ENTER / ENTER / T / FD / W**

# Formatage de la partition

```
mkfs.ext4 /dev/sda
```

# Ajout de la persistence du point de montage

Afin de rendre persistent le montage de notre partition sur le système, nous devons modifier le fichier /etc/fstab pour y ajouter l'entrée due à notre partition :

```
/dev/sda1    /mnt/myPart    ext4    defaults    0    0
```

On peut lancer la commande suivante pour vérifier qu'il n'y a pas d'erreur de syntaxe :

```
mount -a
```

Et on peut recharger la configuration du fichier pour monter notre partition :

```
systemctl daemon-reload
```