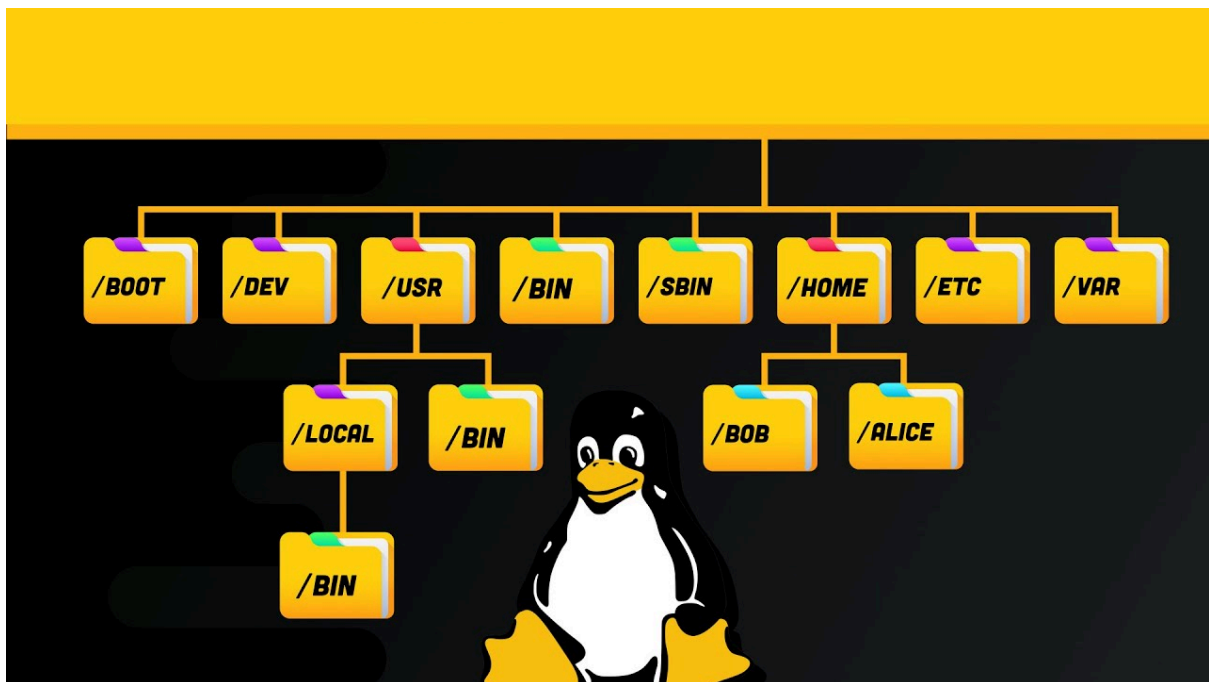


# Manipulation de LINUX File System



# Sommaire

## Prérequis

<b>TP 1</b> - Création et gestion des partitions sous Linux	2
<b>TP 2</b> - Analyse des partitions et compréhension des secteurs	5
<b>TP 3</b> - Exploration des répertoires système et gestion des périphériques	6
<b>TP 4</b> - Installation et gestion des versions du noyau Linux	8

Ceci est une **documentation** présentant les démarches menées pour réaliser les TP sur les **systèmes de fichier**.

Les TP ont été réalisés sur une machine virtuelle **Lubuntu**.

## Qu'est ce qu'un système de fichier ?

Un **système de fichiers** est une structure gérant la manière dont les données sont écrites, lues et structurées sous forme de fichiers et de dossiers.

### Prérequis :

- Une machine virtuelle Linux (ex: Ubuntu, fedora, ...)

---

## **TP 1 - Création et gestion des partitions sous Linux**

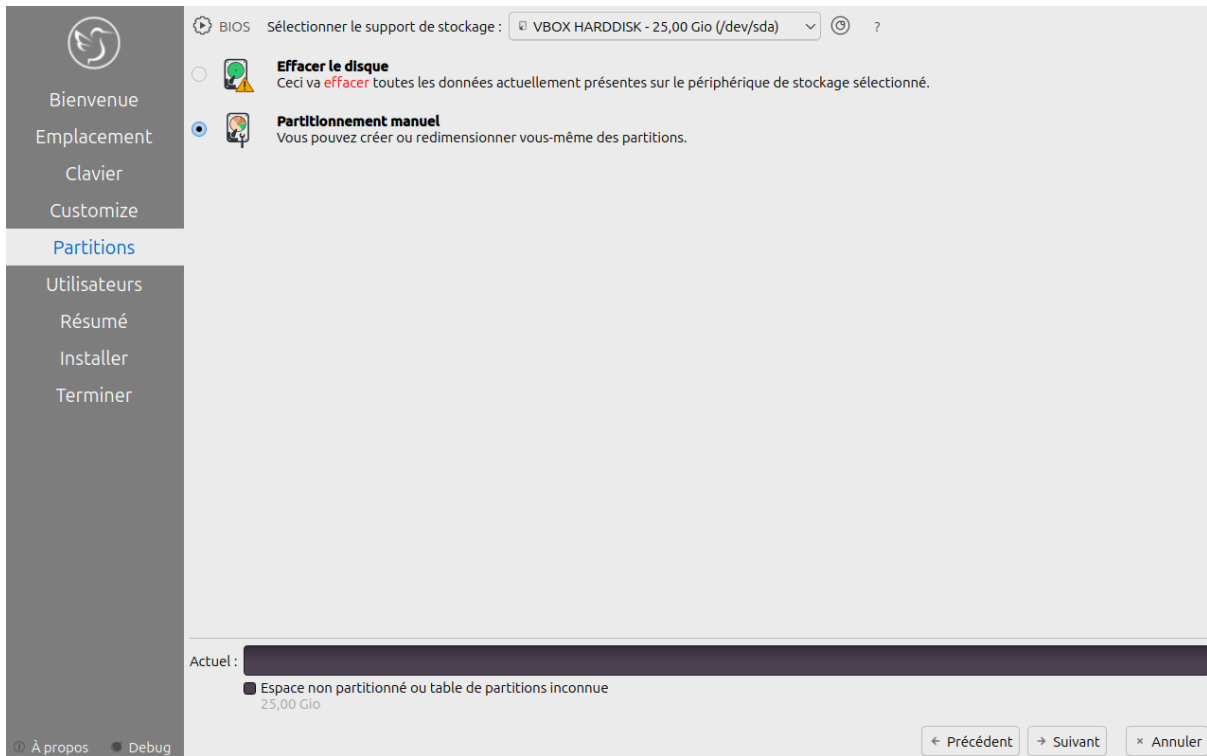
### **Pb:**

Mettez en place un ou plusieurs **système de fichier** (par exemple EXT4, NTFS)... sur votre système linux, puis créer plusieurs **partitions** en utilisant différents points de montages et les associer à des "**flags**".

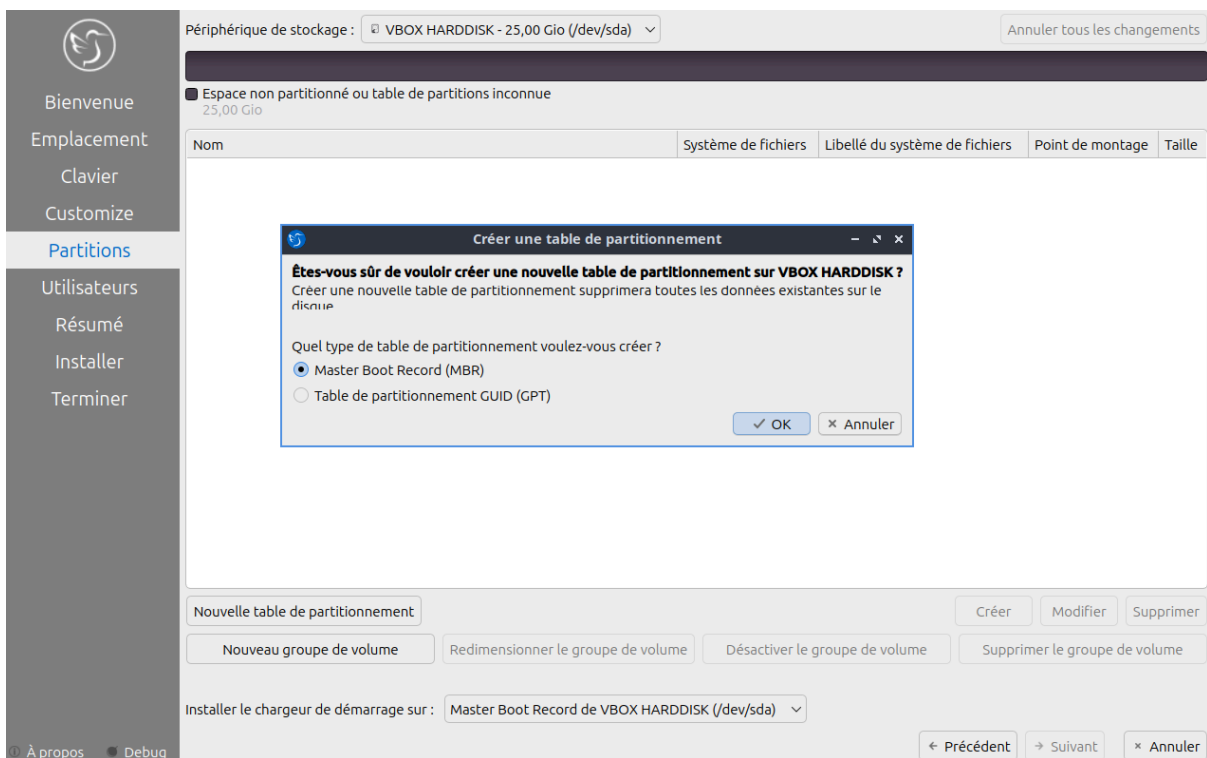
Dans cette documentation j'ai utilisé une machine virtuelle **Lubuntu**, dont l'installation est très simple : il suffit de suivre **8 étapes**.

Pour ce TP nous nous intéressons à une en particulier : **Partitions**.

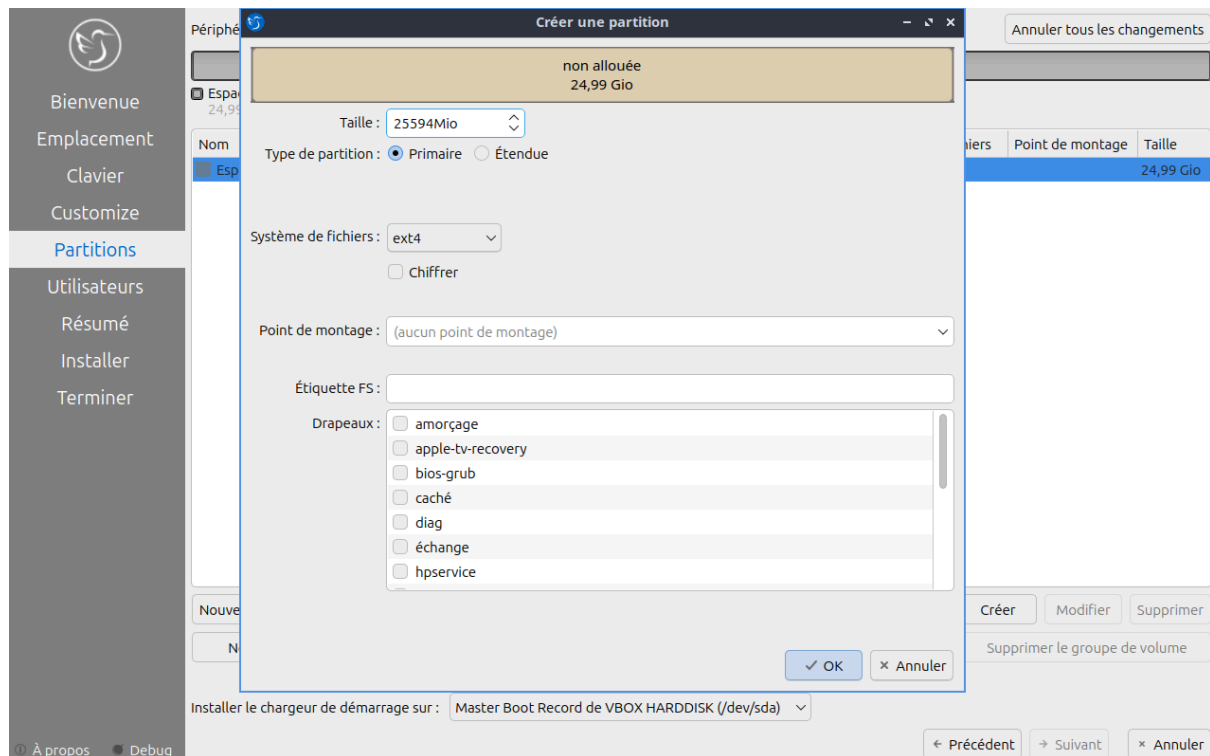
On va sélectionner le partitionnement manuel qui offre un **meilleur contrôle** et **une séparation des données et du système**.



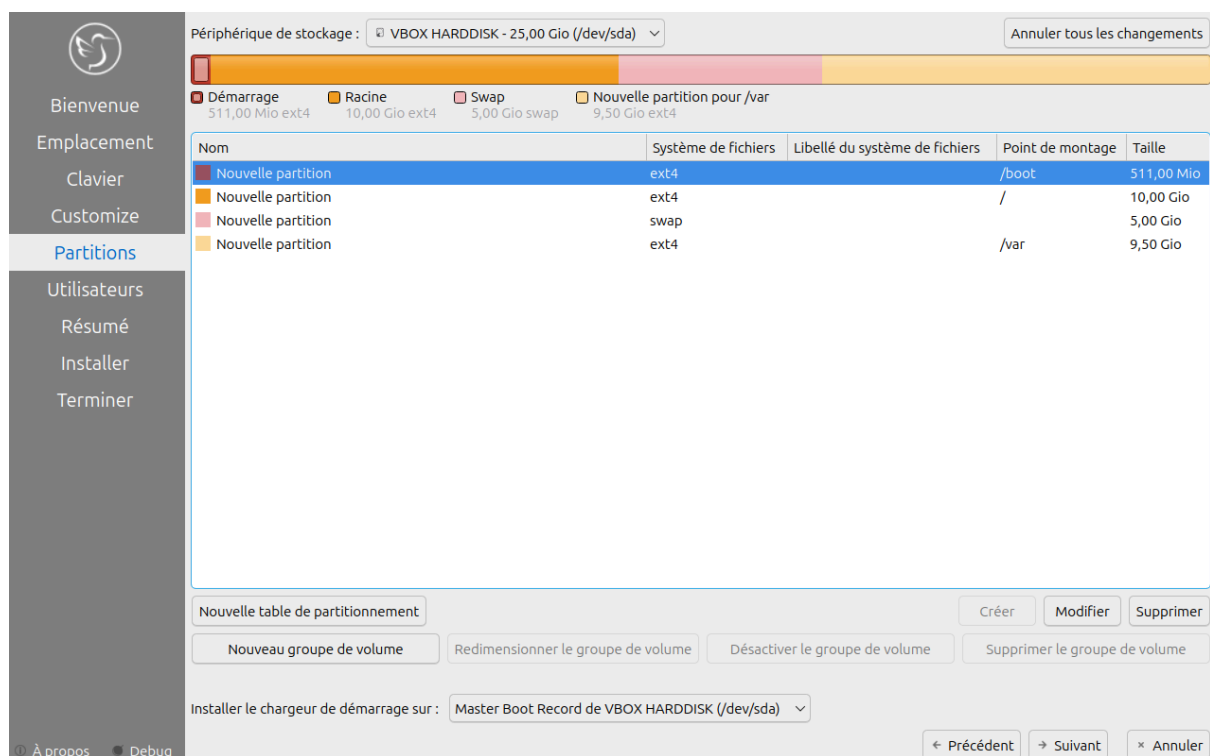
On va créer une nouvelle **table de partitions** en **Master box record** (MBR) :



Puis on va enfin **partitionner** notre disque : ici on va créer **4 partitions**.



Voici notre partitionnement avec un **système de fichier** (ext4) et les **flags** :



Les flags sont : Démarrage, Racine, Swap et Nouvelle partition pour /var

## **TP 2 - Analyse des partitions et compréhension des secteurs**

### **Pb:**

Dans votre système, trouver les informations sur vos partitions et expliquez à quoi correspond le terme "sector" dans votre système de fichier, tout en donnant un exemple.

Sur le terminal on peut utiliser la commande bash ci-dessus pour afficher les partitions :

```
sudo fdisk -l
```

Shell

Ce qui nous donne :

Périphérique	Amorçage	Début	Fin	Secteurs	Taille	Id	Type
/dev/sda1		1048576	22020095	20971520	10G	83	Linux
/dev/sda2		22020096	32505855	10485760	5G	82	partition d'é
/dev/sda3	*	2048	1048575	1046528	511M	83	Linux
/dev/sda4		32505856	52420094	19914239	9,5G	83	Linux

Le terme **sector** ou secteur sur la capture d'écran correspond à la plus petite unité de stockage d'un disque dur.

Chaque partition occupe un certain nombre de secteurs, qui sont référencés par leur numéro de début et de fin.

### **Exemple :**

Pour la partition **/dev/sda1** :

Début : **1048576**

Fin : **22020095**

Nombre de secteurs : **20971520**

Taille : **10G**

Cette **partition** commence au secteur **1 048 576** et s'étend jusqu'au secteur **22 020 095**, occupant **20 971 520** secteurs.

---

## **TP 3 - Exploration des répertoires système et gestion des périphériques**

### **Pb:**

1. Dans quel dossier sous linux on peut trouver des fichiers de bibliothèque partagés et parfois d'autres fichiers liés au noyau ?

**/lib** => contient les fichiers de bibliothèques partagées utilisées par le système et les programmes de base.

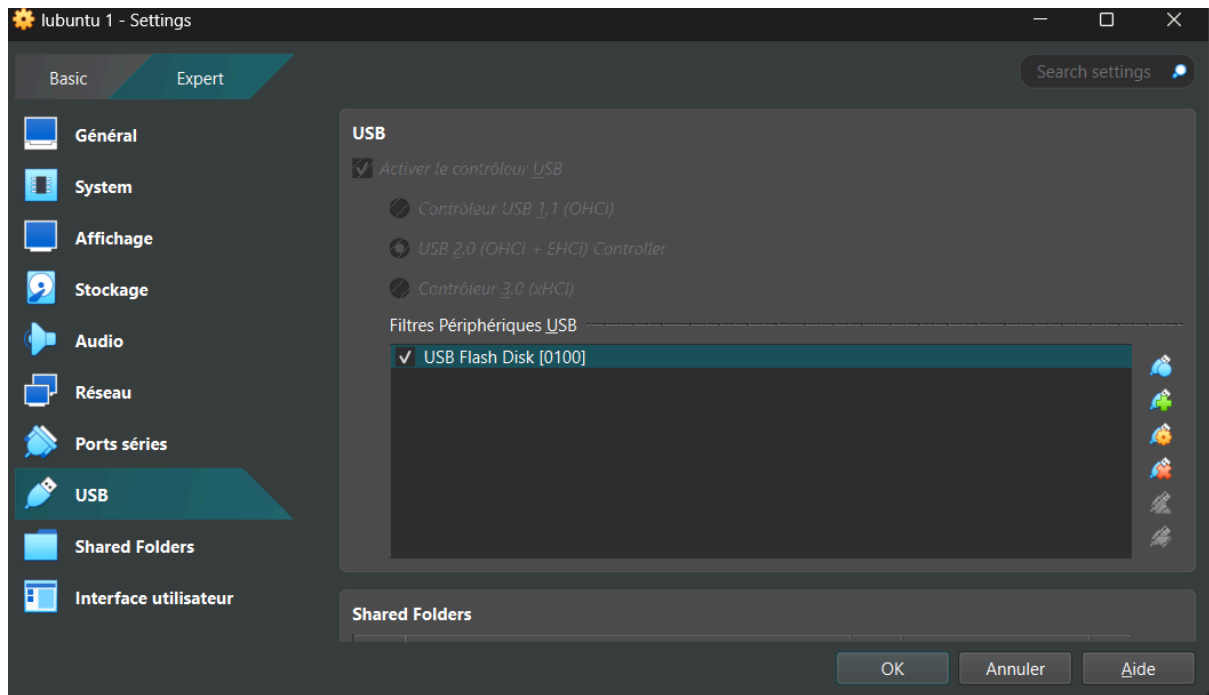
2. Quelles types d'informations pouvons-nous trouver dans le dossier /dev ?

**/dev** => contient les fichiers représentant les périphériques physiques et virtuels du système.

3. Branchez ou montez un périphérique externe physique ou virtuel, puis faites le afficher dans votre système Linux depuis votre système Windows ou MacOS ou encore Linux et dites comment vous avez fait.

Pour avoir la **clé usb** sur la **machine virtuelle**, il suffit d'aller sur **Virtualbox** et suivre les étapes suivantes :

- Aller dans les **paramètres** de la machine
- Se rendre sur **USB**
- Ajouter dans Filtre Périphériques USB : **USB Flash Disk**



Sur **Lubuntu**, pour détecter le **périphérique** (clé-usb pour cette documentation), il suffit d'inscrire la commande **bash** suivante :

```
sudo fdisk -l
```

Shell

Le **terminal** affiche :

```
Disque /dev/sdb : 14,94 GiB, 16039018496 octets, 31326208 secteurs
Disk model: Flash Disk
Unités : secteur de 1 x 512 = 512 octets
Taille de secteur (logique / physique) : 512 octets / 512 octets
taille d'E/S (minimale / optimale) : 512 octets / 512 octets
Type d'étiquette de disque : dos
Identifiant de disque : 0x0015a9a7

Périphérique Amorçage Début      Fin Secteurs Taille Id Type
/dev/sdb1      *          2048 31326143 31324096  14,9G  c W95 FAT32 (LBA)
```

Puis, on crée un **point de montage** :

```
sudo mkdir /mnt/usb
```

Shell



Et enfin, on monte le **périphérique** sur le **répertoire** créé précédemment (le **point de montage**).

```
sudo mount /dev/sdb1 /mnt/usb
```

Shell

Cette manœuvre permet d'afficher le **contenu du périphérique** (clé-usb) sur le **terminal**.

Pour vérifier si elle à bien fonctionnée, il suffit d'inscrire la commande :

```
ls /mnt/usb
```

Shell

Le terminal affiche donc le **contenu** de la **clé-usb** :

```
fch@fch-virtualbox:~$ ls /mnt/usb  
'System Volume Information'
```

---

## **TP 4 - Installation et gestion des versions du noyau Linux**

### **Pb:**

Installer une autre version de votre Kernel linux puis changer l'ordre de démarrage du noyau et faites démarrer votre noyau toujours sur l'ancienne version de votre kernel.

Attention le but ce n'est pas de le faire manuellement au démarrage mais de le rendre automatique.

Pour installer une **autre** version du **Kernel**, il faut d'abord laquelle est installée sur notre machine.

Pour cela on va utiliser la **commande bash** :

```
uname -r
```

Le terminal va renvoyer le nom de la version, ici c'est :

```
fch@fch-virtualbox:~$ uname -r
6.11.0-8-generic
```

Ensuite il faut **vérifier** quelle sont les **autres version disponibles** :

```
apt list --all-versions linux-image-*
```

Ce qui nous affiche une **longue liste de versions** :

```
linux-image-unsigned-6.11.0-18-generic/oracular-updates,oracular-security 6.11.0-18.18 amd64
linux-image-unsigned-6.11.0-19-generic/oracular-updates,oracular-security 6.11.0-19.19 amd64
linux-image-unsigned-6.11.0-8-generic/oracular 6.11.0-8.8 amd64
linux-image-unsigned-6.11.0-9-generic/oracular-updates 6.11.0-9.9 amd64
linux-image-virtual-hwe-24.04-edge/oracular-updates,oracular-security 6.11.0-19.19 amd64
linux-image-virtual-hwe-24.04-edge/oracular 6.11.0-8.8 amd64
linux-image-virtual-hwe-24.04/oracular-updates,oracular-security 6.11.0-19.19 amd64
linux-image-virtual-hwe-24.04/oracular 6.11.0-8.8 amd64
linux-image-virtual/oracular-updates,oracular-security 6.11.0-19.19 amd64
linux-image-virtual/oracular 6.11.0-8.8 amd64
```

Pour ce TP je vais utiliser la version **6.11.0-1001-realtime**.

Ensuite on va **installer** la nouvelle version du **noyau** :

```
sudo apt update
```

Shell

```
sudo apt install linux-image-6.11.0-1001-realtime
```

Shell

Pour **appliquer** l'installation du **noyau** sur le **système d'exploitation**, il est nécessaire de faire des réglages sur **GRUB** (GRand Unified Bootloader).

D'abord, on va mettre à jour **GRUB** :

```
sudo update-grub
```

Shell

Ensuite, on va lister les noyaux disponibles sur GRUB, soit ceux que l'on a installés précédemment :

```
sudo grep -i 'menuentry' /boot/grub/grub.cfg
```

Shell

```
fch@fch-virtualbox:~$ sudo grep -i 'menuentry' /boot/grub/grub.cfg
if [ x"${feature_menuentry_id}" = xy ]; then
  menuentry_id_option="--id"
  menuentry_id_option=""
export menuentry_id_option
menuentry 'Ubuntu' --class ubuntu --class gnu-linux --class gnu --class
os $menuentry_id_option 'gnulinux-simple-653bd796-0b40-42db-b08c-97e7f01
3f649' {
submenu 'Advanced options for Ubuntu' $menuentry_id_option 'gnulinux-adv
anced-653bd796-0b40-42db-b08c-97e7f013f649' {
  menuentry 'Ubuntu, with Linux 6.11.0-1003-realtime' --class ubun
tu --class gnu-linux --class gnu --class os $menuentry_id_option 'gnulin
ux-6.11.0-1003-realtime-advanced-653bd796-0b40-42db-b08c-97e7f013f649' {
    menuentry 'Ubuntu, with Linux 6.11.0-1003-realtime (recovery mod
e)' --class ubuntu --class gnu-linux --class gnu --class os $menuentry_i
d_option 'gnulinux-6.11.0-1003-realtime-recovery-653bd796-0b40-42db-b08c
-97e7f013f649' {
      menuentry 'Ubuntu, with Linux 6.11.0-1001-realtime' --class ubun
tu --class gnu-linux --class gnu --class os $menuentry_id_option 'gnulin
ux-6.11.0-1001-realtime-advanced-653bd796-0b40-42db-b08c-97e7f013f649' {
        menuentry 'Ubuntu, with Linux 6.11.0-1001-realtime (recovery mod
e)' --class ubuntu --class gnu-linux --class gnu --class os $menuentry_i
d_option 'gnulinux-6.11.0-1001-realtime-recovery-653bd796-0b40-42db-b08c
```

On veut **éditer** le fichier de configuration de GRUB pour démarrer automatiquement sur l'**ancienne version** :

```
sudo nano /etc/default/grub
```

Shell

Dans le fichier, on va changer la variable **GRUB\_DEFAULT** pour passer de **0** à **"1>2"** :

```
GNU nano 8.1 /etc/default/grub *
# If you change this file or any /etc/default/grub.d/*.cfg file,
# run 'update-grub' afterwards to update /boot/grub/grub.cfg.
# For full documentation of the options in these files, see:
#   info -f grub -n 'Simple configuration'

GRUB_DEFAULT=0
GRUB_TIMEOUT_STYLE=hidden
GRUB_TIMEOUT=0
GRUB_DISTRIBUTOR='Ubuntu'
GRUB_CMDLINE_LINUX_DEFAULT='quiet splash resume=UUID=b8ba1c80-7cba-485a>
GRUB_CMDLINE_LINUX=""

# If your computer has multiple operating systems installed, then you
# probably want to run os-prober. However, if your computer is a host
# for guest OSes installed via LVM or raw disk devices, running
# os-prober can cause damage to those guest OSes as it mounts
# filesystems to look for things.
#GRUB_DISABLE_OS_PROBER=false

# Uncomment to enable BadRAM filtering, modify to suit your needs

GNU nano 8.1 /etc/default/grub *
# If you change this file or any /etc/default/grub.d/*.cfg file,
# run 'update-grub' afterwards to update /boot/grub/grub.cfg.
# For full documentation of the options in these files, see:
#   info -f grub -n 'Simple configuration'

GRUB_DEFAULT="1>2"
GRUB_TIMEOUT_STYLE=hidden
GRUB_TIMEOUT=0
GRUB_DISTRIBUTOR='Ubuntu'
GRUB_CMDLINE_LINUX_DEFAULT='quiet splash resume=UUID=b8ba1c80-7cba-485a>
GRUB_CMDLINE_LINUX=""

# If your computer has multiple operating systems installed, then you
# probably want to run os-prober. However, if your computer is a host
# for guest OSes installed via LVM or raw disk devices, running
# os-prober can cause damage to those guest OSes as it mounts
# filesystems to look for things.
#GRUB_DISABLE_OS_PROBER=false

# Uncomment to enable BadRAM filtering, modify to suit your needs
```

Pour **appliquer** les **modifications**, il faut écrire les commandes suivantes :

```
sudo update-grub
```

Shell

```
reboot
```

Shell

Après le **redémarrage** du système d'exploitation (OS), on peut vérifier si la **version du noyau** a changé :

```
uname -r
```

Shell

```
fch@fch-virtualbox:~$ uname -r  
6.11.0-1001-realtime
```

Si la **version** a changé, c'est que **ça fonctionne !**