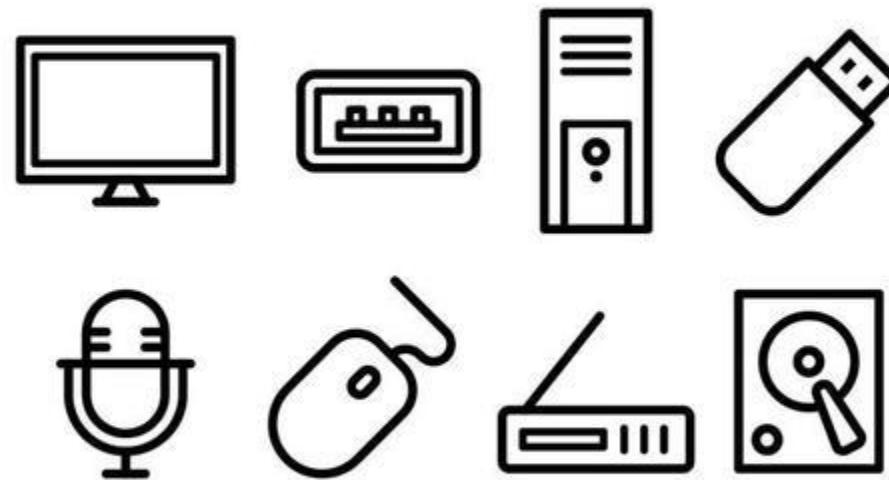


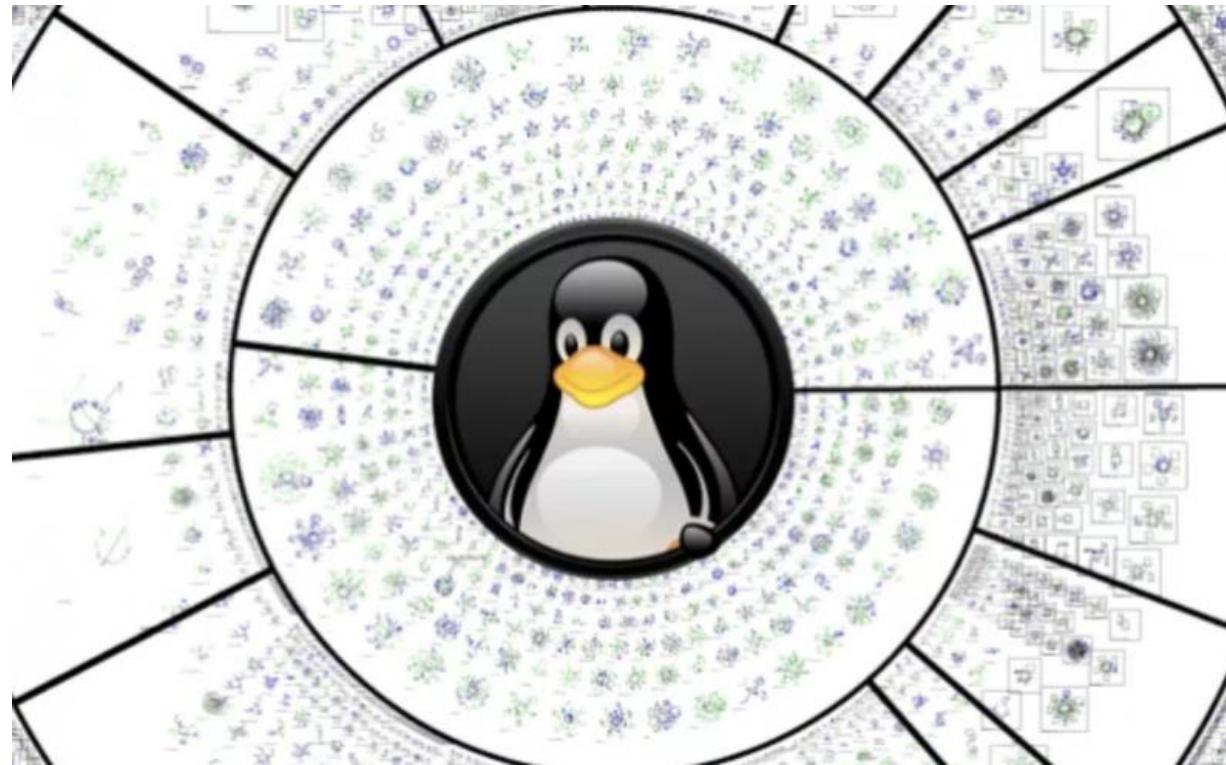
Configuration matérielle: périphériques et disques

Partie 1: Gestion des périphériques



Le noyau

- Le noyau est le cœur du système
- S'occupe de fournir aux logiciels une interface pour utiliser le matériel



Les fichiers du noyau

Les fichiers de démarrage du système se trouvant dans /boot

- fichier initrd est une image d'un système d'exploitation minimal initialisé après démarrage du système
- fichier system.map contient une table avec les symboles et leur adresse mémoire
- fichier /boot/vmlinuz-* est le noyau Linux compressé qui sera utilisé après démarrage
- un fichier de configuration du noyau est présent dans le répertoire boot: /boot/config

La représentation des périphériques pour le noyau

- Les périphériques sont représentés par des noms de fichiers, et peuvent être utilisés comme des fichiers ordinaires
- /dev contient les fichiers associés aux périphériques:
- Ex:
- lp : imprimante
- ttyn : terminal n°n
- hkn : disque dur n°n
- mtn : bande magnétique n°n
- on liste le contenu de /dev: ls -l /dev
- exemple pour les disques dur: ls -l /dev/sd*

Le système de fichier virtuel **udev**

- Linux dispose d'une fonctionnalité permettant aux applications de détecter l'apparition et la suppression des périphériques
- cette fonctionnalité est principalement utilisée par le sous système udev(abréviation de Userspace /dev) comme son nom l'indique, udev a pour principale fonction de prendre en charge la gestion du répertoire /dev/



Le système de fichier virtuel **udev**

- **udev** est un élément clé de la gestion matérielle sous Linux. Il agit comme un **gestionnaire dynamique de périphériques**, permettant la détection et la configuration automatique des matériels dès qu'ils sont connectés ou déconnectés. C'est lui qui crée ou supprime les fichiers dans le répertoire `/dev` en temps réel.
- Fonctionnement de udev:
Événement matériel → Noyau → udev → Création du fichier `/dev`

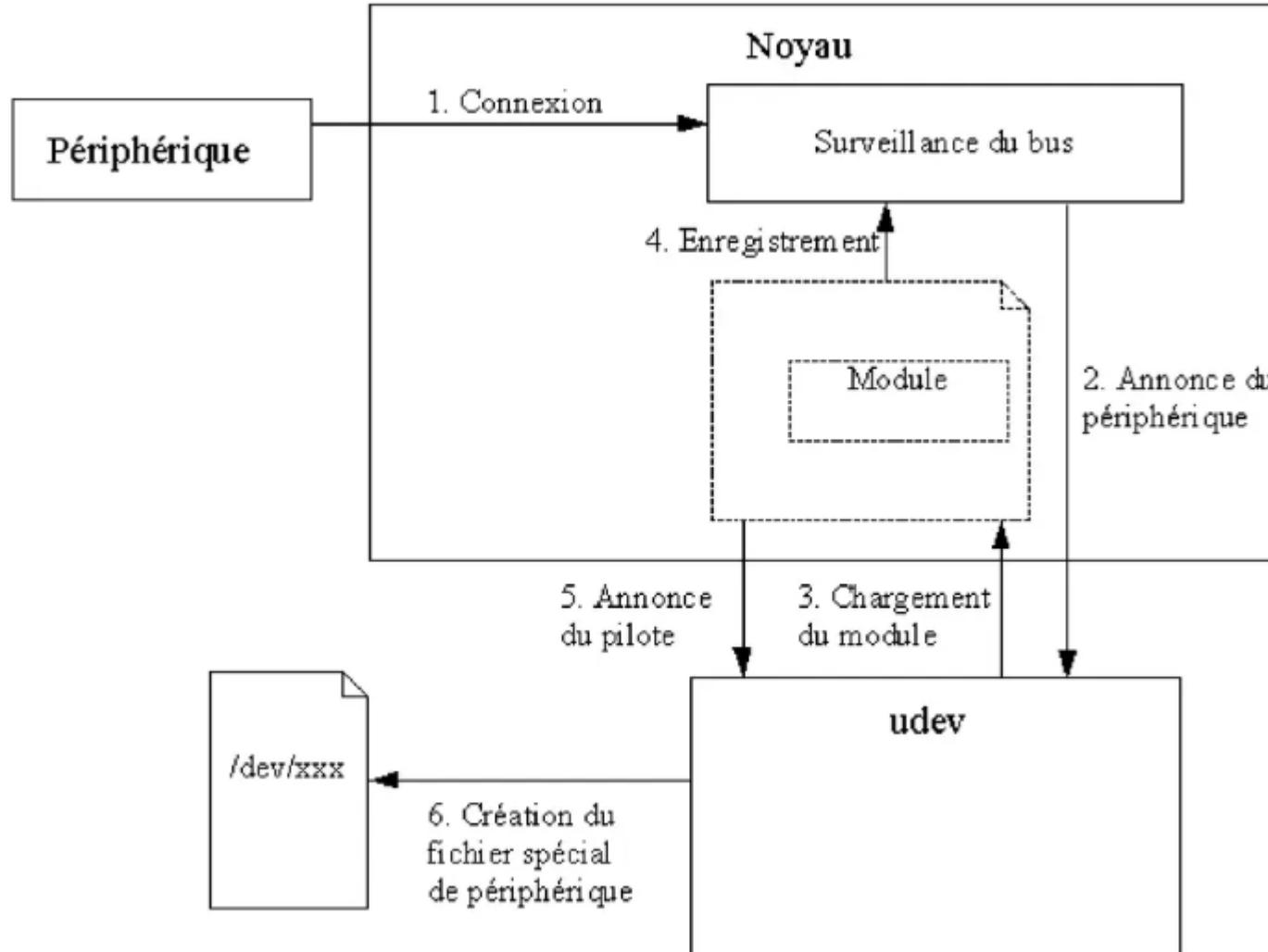
La détection automatique du matériel

udev peut réaliser les opérations suivantes lorsque le noyau signale la présence d'un nouveau périphérique:

- chargement du module du pilote de périphérique si nécessaire
- si le périphérique requiert un firmware chargement de celui ci
- création du fichier spécial de périphérique et de ses alias nécessaires a l'utilisation du périphérique
- exécution des opérations d'initialisation complémentes ou lancement des applications utilitaires associés au périphérique
- notification de la présence du périphérique a l'ensemble des autres programmes qui s'intéressent a la gestion du matériel (par exemple le gestionnaire de bureau)



Création des fichiers spéciaux de périphériques



Règles udev



- Le comportement de **udev** est contrôlé par des **règles**, situées principalement dans les dossiers :
 - /etc/udev/rules.d/
 - /lib/udev/rules.d/
- Chaque règle définit comment **udev** doit gérer un périphérique : nommage personnalisé, permissions spécifiques, ou lancement de scripts.
- Exemple simple de règle udev (dans /etc/udev/rules.d/10-usb.rules) :
 - ACTION=="add", SUBSYSTEM=="usb", ATTR{idVendor}=="abcd", ATTR{idProduct}=="1234", SYMLINK+="mon_usb"
- Cette règle crée un lien symbolique /dev/mon_usb pour tout périphérique USB correspondant à l'ID abcd:1234.

Commandes udevadm

- **Surveiller les événements matériels en temps réel**
- Pour voir en direct ce qui se passe lorsque vous connectez ou déconnectez un périphérique :
- `udevadm monitor`
- Cette commande affiche les événements du noyau et les actions prises par `udev` :
 - KERNEL[1534.5678] add /devices/pci0000:00/0000:00:1d.0/usb2/2-1 (usb)
 - UDEV [1534.5690] add /devices/pci0000:00/0000:00:1d.0/usb2/2-1 (usb)

Commandes udevadm

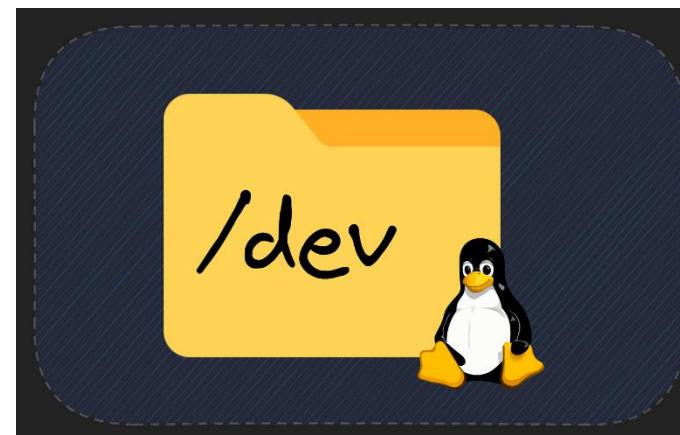
- **Lister les informations sur un périphérique**
- Pour obtenir toutes les propriétés d'un périphérique (par exemple /dev/sdb1) :
- `udevadm info --name=/dev/sdb1`
- Cela affichera des informations détaillées sur le périphérique, y compris son numéro de série, son type, et d'autres attributs.

Commandes udevadm

- **Recharger les règles udev**
- Après avoir modifié une règle, vous pouvez demander à **udev** de recharger ses règles avec :
- `sudo udevadm control --reload`
- Et pour appliquer immédiatement ces règles à un périphérique déjà branché :
- `sudo udevadm trigger`

Les fichiers de périphériques dans /dev

- Le répertoire `/dev` contient des **fichiers de périphériques** qui servent d'interface entre le noyau et le matériel. Lorsqu'une application souhaite interagir avec un périphérique (par exemple, lire un fichier sur un disque), elle passe par ce fichier spécial.
- Exemple de commande pour lister les périphériques :
- `ls -l /dev`



Les fichiers de périphériques dans /dev

Il existe deux grandes catégories de fichiers de périphériques :

- **Périphériques de caractères (type c)** Ils transmettent les données **caractère par caractère**. C'est le cas des terminaux (/dev/tty), des ports série ou des périphériques audio.
- **Périphériques de blocs (type b)** Ils gèrent les données **en blocs**, ce qui est plus efficace pour les disques durs, les clés USB ou les partitions (/dev/sda, /dev/sdb1).

Chaque fichier est identifié par deux numéros :

- **Numéro majeur** : identifie le type de périphérique (le pilote utilisé).
- **Numéro mineur** : identifie l'instance spécifique du périphérique.
- Vous pouvez consulter ces numéros avec la commande ls -l, comme vu précédemment.

Informations sur le matériel: lshw

La commande **lshw** (Hardware Lister) est un outil puissant qui permet d'afficher des informations détaillées sur l'ensemble du matériel de votre système : processeur, mémoire, cartes réseau, stockage... Contrairement à lspci ou lsusb, qui se concentrent sur des bus spécifiques, **lshw** donne une vue globale de la machine.

lshw interroge directement le noyau et les interfaces systèmes pour récupérer des données techniques précises, notamment sur :

- La carte mère
- Le processeur (CPU)
- La mémoire (RAM)
- Les périphériques de stockage (HDD, SSD)
- Les interfaces réseau

Pour afficher **toutes les informations** disponibles sur votre matériel :

- sudo lshw



Informations sur le matériel: lshw

Vous pouvez filtrer les résultats pour n'afficher qu'une catégorie précise de périphériques :

Pour les cartes réseau :

- sudo lshw -C network

Pour le stockage :

- sudo lshw -C storage

Pour la mémoire :

- sudo lshw -C memory

Vous pouvez exporter les résultats en XML, HTML ou JSON pour les intégrer dans des rapports ou des outils externes.

- Format HTML :
- sudo lshw -html > rapport.hardware.html
 - Format JSON (selon la version) :
- sudo lshw -json



Identifier les périphériques PCI avec lspci

- Le bus **PCI (Peripheral Component Interconnect)** est utilisé pour connecter une grande variété de périphériques matériels internes : cartes réseau, cartes graphiques, contrôleurs SATA, etc.
- La commande **lspci** permet d'identifier tous les périphériques connectés à ce bus, ce qui est particulièrement utile pour diagnostiquer ou documenter la configuration matérielle d'une machine.
- Pour lister tous les périphériques PCI présents sur votre machine :
- `lspci`
- Pour obtenir des informations plus précises sur chaque périphérique, utilisez l'option `-v` (verbose) :
- `lspci -v`
- Pour afficher les identifiants numériques du fabricant et du périphérique, ajouter l'option `-nn`



Identifier les périphériques USB avec lsusb

- Le bus **USB (Universal Serial Bus)** est omniprésent : clés USB, disques externes, claviers, souris, webcams... Tous ces périphériques peuvent être listés et identifiés avec la commande **lsusb**. Cet outil est indispensable pour diagnostiquer des problèmes liés aux périphériques externes ou vérifier leur bonne détection par le système.
- Pour afficher la liste des périphériques USB connectés :
- lsusb



Identifier les périphériques USB avec lsusb

- Pour obtenir des informations complètes sur chaque périphérique (interfaces, protocoles, alimentations...), utilisez :
- `lsusb -v`
- Attention, cette commande peut générer beaucoup d'informations si plusieurs périphériques sont branchés.
- Pour cibler un seul périphérique (par exemple Bus 002 Device 003), vous pouvez préciser l'adresse :
- `lsusb -s 002:003 -v`
- Cela vous donnera tous les détails techniques uniquement pour ce périphérique.
- Pour afficher l'arborescence des périphériques USB (utile pour voir comment chaque périphérique est connecté) :
- `lsusb -t`

Visualiser la hiérarchie des périphériques de blocs avec lsblk

La commande **lsblk** (List Block Devices) permet d'afficher la hiérarchie complète des **périphériques de bloc** : disques durs, SSD, clés USB, partitions, etc. Elle est particulièrement utile pour vérifier la structure des disques et des systèmes de fichiers.

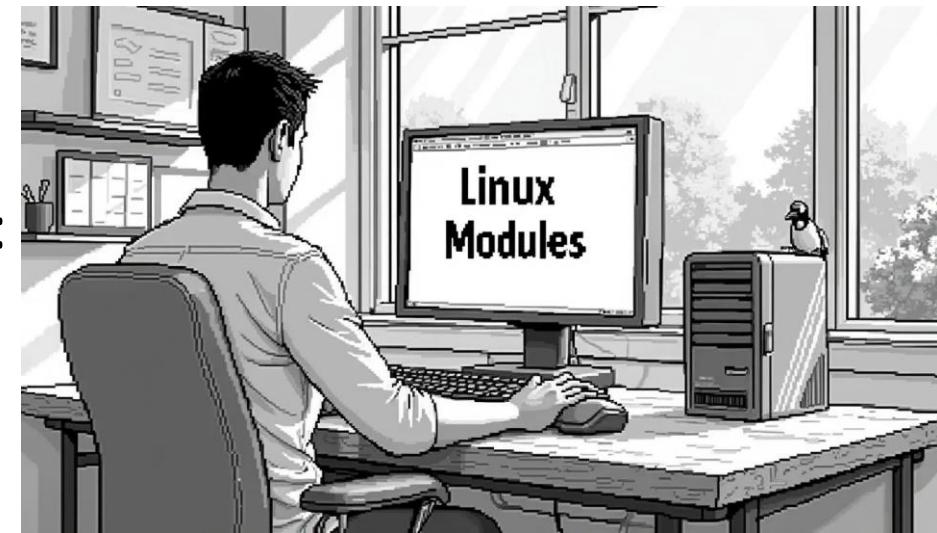
Pour lister tous les périphériques de bloc détectés par le système :

- lsblk



Gestion des modules du noyau avec modprobe et lsmod

- Le **noyau Linux** utilise des **modules** pour gérer de nombreux composants matériels. Ces modules sont des morceaux de code qui peuvent être chargés ou déchargés dynamiquement, permettant au noyau de **s'adapter en temps réel** aux périphériques présents sur la machine.
- La commande **lsmod** affiche tous les modules actuellement chargés dans le noyau.
- **lsmod**
- Pour charger un module spécifique (par exemple `usb-storage` pour un périphérique USB de stockage) :
`sudo modprobe usb-storage`



Gestion des modules du noyau avec modprobe et lsmod

- Pour vérifier que le module a bien été chargé ou déchargé, combinez :
- lsmod | grep usb_storage
- La commande **modinfo** permet d'obtenir des détails sur un module : description, licence, dépendances, alias matériels...
- modinfo usb-storage



Partie 2: Gestion du disque



Gestion du système fichiers et disques

- La gestion des systèmes de fichiers et des disques constitue une **composante essentielle de l'administration système**. Elle permet d'organiser, stocker et accéder efficacement aux données sur les supports physiques ou virtuels.
- Utiliser les commandes et les concepts essentiels liés à la gestion de l'espace disque



Affichage de l'utilisation du disque avec df

- La commande df (disk free, espace disque libre) est l'outil idéal pour vérifier l'utilisation de l'espace disque sur votre système Linux.
Explorons comment l'utiliser :
- Ouvrez votre terminal. Vous devriez être dans le répertoire /home/user/Documents. Si vous n'êtes pas sûr, vous pouvez toujours vérifier votre répertoire actuel avec la commande pwd.
- Exédez la commande suivante pour afficher l'utilisation du disque :
- df



Gestion des partitions de disque avec fdisk

- **Partitions** : Une partition est une section d'un disque qui est traitée comme une unité distincte. Pensez-y comme à diviser un grand disque dur en sections plus petites et indépendantes



Gestion des partitions de disque avec fdisk

- Dans un système réel, avant de pouvoir créer un système de fichiers, vous avez souvent besoin de créer des partitions. Bien que nous ne puissions pas modifier les partitions de disque réelles dans cet environnement virtuel, nous pouvons explorer comment utiliser fdisk pour afficher les informations sur les partitions.
- Tout d'abord, affichons les informations sur toutes les partitions de disque :
- `sudo fdisk -l`



Gestion des partitions de disque avec fdisk

- Une fois que vous avez trouver **le nom du périphérique du disque**, on l'indique dans **la commande fdisk** pour travailler avec.
- `fdisk /dev/sdb`
- Les commandes et actions se font en saisissant une lettre.
Pour afficher la liste des lettre et commandes, **appuyez sur m** et validez.
Vous établissez vos modifications et vous les **appliquez avec w**, ce qui va **écrire sur le nouveau partitionnement sur le disque**.

Gestion des partitions de disque avec fdisk

- Voici la liste des commandes de fdisk les plus utiles :

Lettre	Description
d	supprimer la partition
F	afficher l'espace libre non partitionné
I	afficher les types de partitions connues
n	ajouter une nouvelle partition
p	afficher la table de partitions

Lettre	Description
t	modifier le type d'une partition
v	vérifier la table de partitions
i	Afficher des renseignements sur la partition
w	écrire la table sur le disque et quitter
q	quitter sans enregistrer les modifications

Créer un système de fichiers

- La commande mkfs (make filesystem) est utilisée sous Linux pour **créer un système de fichiers** sur une partition ou un disque.
- Elle **formate** une partition en y installant un système de fichiers comme :
- ext4 (le plus courant sous Linux)
- xfs, btrfs, vfat, ntfs, etc.

Syntaxe :

- `mkfs -t ext4 /dev/sdXn`

Montage

- Sous **Linux**, le système repose sur une arborescence unique commençant à la racine /. Tous les **disques** et **partitions** doivent être **montés** sur des **points de montage** pour devenir visibles et utilisables par le système.
- Un **point de montage** est simplement un **répertoire** où un **système de fichiers** externe est attaché à l'arborescence. Ce répertoire peut être un dossier vide existant, comme /mnt, /media/usb, ou n'importe quel autre répertoire personnalisé.

Monter manuellement une partition

- Monter une **partition** manuellement sous **Linux** permet de rendre immédiatement un **système de fichiers** accessible sans avoir à redémarrer ni modifier la configuration permanente.
- **Utiliser la commande mount**
- Pour monter une partition, utilisez la commande suivante :
- `sudo mount /dev/sdXN /mnt`
- **Exemple** concret :
- `sudo mount /dev/sdb1 /mnt/backup`
- Cela permet d'accéder aux fichiers de `/dev/sdb1` à partir du répertoire `/mnt/backup`.

Vérification du montage

- Après avoir monté une partition, on peut vérifier que l'opération s'est déroulée correctement.
- **Lister les systèmes de fichiers montés :**
- `mount | grep /dev/sdb1`

Démontage d'un système de fichiers

- Avant de déconnecter un **disque** ou de retirer une **clé USB** sous **Linux**, il est essentiel de **démonter** proprement la partition montée. Cela garantit l'intégrité des données en assurant que toutes les écritures en cours sont terminées.
- **Utiliser umount proprement**
- Pour démonter une partition montée, utilisez la commande suivante :
- `sudo umount /mnt/backup`
- ou, en indiquant directement le périphérique :
- `sudo umount /dev/sdb1`

TP 1: Gestion des périphériques

- 1. Lister les composants internes connectés via le bus PCI (carte réseau, graphique, etc.)
- 2. Afficher les périphériques USB connectés (même s'il n'y en a pas)
- 3. Affiche une liste complète des périphériques d'entrée reconnus par le noyau Linux avec cat /proc/bus/input/devices
- 4. affiche les périphériques de sortie de son avec aplay –l et arecord –l
- 5. affiche les périphériques de sortie vidéo et la résolution d'écran avec xrandr
- 6. affiche l'imprimante avec lpstat -p -d

TP 2: Créer disque virtuel

1.Tout d'abord, créons un disque virtuel de 256 Mo en utilisant la commande dd :

- dd if=/dev/zero of=virtual.img bs=1M count=256

2. Vérifions la taille du fichier :

- ls -lh virtual.img Vous devriez voir que virtual.img fait exactement 256 Mo.

3.Maintenant, formatons ce disque virtuel avec un système de fichiers ext4 :

- sudo mkfs.ext4 virtual.img



TP 2: Créer disque virtuel

4. Ensuite, on créer un point de montage. C'est le répertoire où le contenu du disque virtuel apparaîtra :

- sudo mkdir /mnt/virtualdisk

5. Monter le disque virtuel :

- sudo mount -o loop virtual.img /mnt/virtualdisk

6. Vérifier que le disque est monté :

- mount | grep virtualdisk

7. Maintenant qu'il est monté, nous pouvons l'utiliser comme n'importe quel autre répertoire. Créons un fichier :

- sudo touch /mnt/virtualdisk/testfile ls /mnt/virtualdisk

8. Lorsque vous avez terminé d'utiliser le disque virtuel, vous devez le démonter :

- sudo umount /mnt/virtualdisk



TP 3: Création de 3 partitions

1. Ouvrir fdisk sur le disque cible
2. Créer la première partition
 - Taper n → nouvelle partition
 - Choisissez p → partition principale
 - Sélectionner le numéro (ex. : 1)
 - Définir le début et la fin
3. Créer la deuxième partition
 - Répéter : n, p, 2...
4. Créer la troisième partition
 - Répéter : n, p, 3..
5. Vérifier la table des partitions: p
6. Enregistrer les modifications : w

