

# Gestion des données et logiciels

# Manipulation de fichiers



## Commandes de manipulation de texte:

- **Cut:** utilisée pour extraire des portions spécifiques de lignes de texte à partir de fichiers ou de flux de données.
- **Sort:** utilisée pour trier les lignes d'un fichier ou de la sortie d'une commande en fonction de critères de tri spécifiés.
- **Uniq:** nettoyer les données en éliminant les doublons
- **Tr:** traduire, de supprimer ou de remplacer des caractères spécifiques dans le texte.
- **wc (word count):** compter le nombre de lignes, de mots et d'octets (caractères) dans un fichier
- **Tail:** permet d'afficher les dernières lignes d'un fichier texte

# Commandes de manipulation de texte:

**-d → Délimiteur (utilisé avec cut):** Définit le **caractère séparateur** entre les champs d'une ligne

- Exemple : `cut -d ',' -f1 fichier.csv`

→ Utilise la **virgule** comme séparateur

**-f → Champ(s) à extraire (utilisé avec cut):** Spécifie **quel(s) champ(s)** afficher

- Exemple : `cut -d ',' -f1,6 fichier.csv`

→ Affiche les **champs 1 et 6**

**-n → Numéro de ligne (utilisé avec tail):** Permet de **commencer à partir d'une ligne spécifique**

- Exemple : `tail -n +2 fichier.csv`

• → Ignore la **première ligne** (souvent l'en-tête)

# Editeur de texte nano

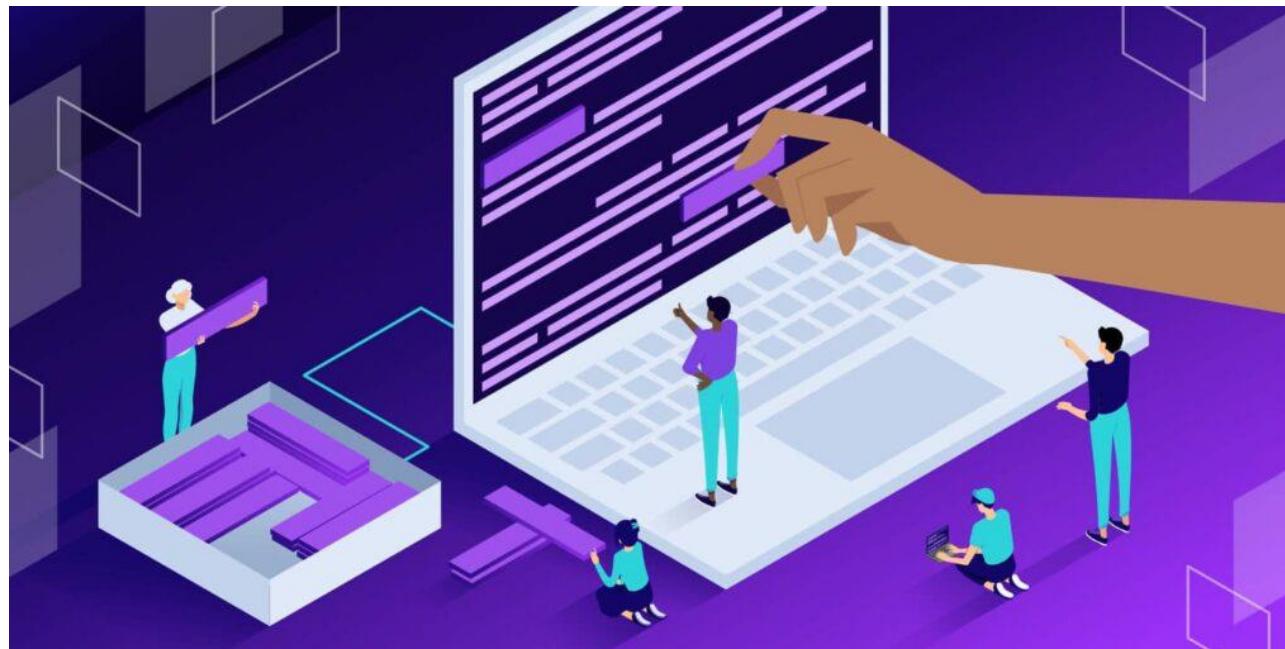
Nano est un éditeur de texte en ligne de commande très pratique pour les utilisateurs de Linux.

- Syntaxe :nano nom\_du\_fichier
- Dans l'éditeur nano, pour enregistrer vos modifications, utilisez Ctrl+O puis Enter. Et pour quitter nano, utilisez Ctrl+X.

# Variables d'environnement

Une variable d'environnement est une **valeur dynamique** utilisée par le système pour configurer le comportement des processus.

- Elle est stockée sous forme de **paires clé=valeur**.



# Variables d'environnement sous Linux

Une variable d'environnement est une **valeur dynamique** utilisée par le système pour configurer le comportement des processus.

- Elle est stockée sous forme de **paires clé=valeur**.

Exemples :

- PATH : liste des répertoires contenant les commandes exécutables
- HOME : répertoire personnel de l'utilisateur
- USER : nom de l'utilisateur connecté
- SHELL : shell par défaut (ex. : /bin/bash)

# Variables d'environnement dans Windows

Une variable d'environnement est une **valeur dynamique** utilisée par le système pour configurer le comportement des processus.

- Elle est stockée sous forme de **paires clé=valeur**.

Exemples :

- PATH : liste des dossiers contenant les exécutables
- TEMP : dossier temporaire
- USERNAME : nom de l'utilisateur connecté
- SystemRoot : dossier d'installation de Windows

# Modifier ou créer une variable temporaire

Sur powershell

- \$env:MON\_VAR = "MaValeur "

Sous shell

- NOM="valeur"
- export NOM

# Créer une variable persistante

- Sous linux:
- Ajouter la ligne dans le fichier de configuration du shell : `~/.bashrc` ou `~/.profile`
- Exemple:
- `echo 'export COURS="Linux"' >> ~/.bashrc`
- `source ~/.bashrc`
  
- Sous windows:
- `setx MON_COURS "Windows"`

# Accès privilégié

- Le mode privilégié est le plus souvent activé en accédant à un **compte d'utilisateur privilégié**. Nous disons donc que sur un système, nous avons des comptes d'utilisateurs réguliers (utilisateurs réguliers) et des comptes d'utilisateurs privilégiés (utilisateur privilégié). L'utilisateur privilégié est aussi appelé administrateur, admin ou superutilisateur. Sous Linux, l'utilisateur privilégié est appelé **root**.

# root

- C'est le **superutilisateur** du système Linux
- Il possède **tous les droits** sur tous les fichiers et processus
- L'accès direct nécessite le **mot de passe root**



/ROOT:#

# Sécurité avec sudo

- La commande sudo (SuperUser DO) permet à des utilisateurs autorisés d'exécuter certaines commandes en tant qu'un autre utilisateur, généralement l'utilisateur root.
- sudo est un moyen efficace d'administrer un système Linux sans avoir à se connecter en tant que root, ce qui peut être risqué.



# Pourquoi utiliser sudo ?

Utiliser sudo présente plusieurs avantages :

- **Sécurité accrue** : En limitant l'accès aux privilèges d'administrateur, on réduit le risque de modifications accidentelles ou malveillantes du système.
- **Auditabilité** : Les actions effectuées avec sudo sont généralement enregistrées dans des journaux, ce qui permet de suivre qui a fait quoi et quand.
- **Contrôle granulaire** : Il est possible de configurer sudo pour permettre à certains utilisateurs d'exécuter uniquement des commandes spécifiques avec des privilèges élevés.
- **Réduction des erreurs** : En utilisant sudo, les utilisateurs sont moins susceptibles de faire des erreurs critiques, car ils n'ont pas un accès complet au système.

# sudo

Pour exécuter une commande en tant que root sans changer d'user : sudo commande.

- Le mot de passe de utilisateur lançant sudo est demandé par le système.
- Seuls certains utilisateurs appelés sudoers ont le droit d'exécuter la commande sudo.

Pour ajouter un utilisateur dans la liste des sudoers, exécuté en tant que root (ou via un autre utilisateur sudoers) :

- adduser user sudo (en tant que root)
- sudo adduser user sudo (avec un utilisateur déjà membre des sudoers)

# sudo

- sudo su est un **raccourci pratique** pour passer root sans connaître son mot de passe.
- sudo su - : il simule une **connexion complète en root** (comme si tu avais ouvert une session root depuis l'écran de connexion).
- Pour les tâches ponctuelles, sudo commande reste la méthode la plus sécurisée.
- **Voir l'historique des commandes sudo** : en utilisant la commande :  
sudo –l
- **Exécuter plusieurs commandes avec sudo** : exemple : sudo sh -c "commande1 ; commande2 ; commande3"

# SU

- Pour se connecter en tant qu'utilisateur souvent root (et pouvoir faire les opérations qui lui sont réservées) : su root.
- Le mot de passe de utilisateur root est demandé par le système.

su et su - donnent un accès prolongé à root, mais avec des différences :

- su : accès sans charger l'environnement (variables, chemin, etc.)
- su - : accès **comme si on s'était connecté directement** en root

Pour agir en tant qu'utilisateur *user*, exécuter :

- su user, ou su - user pour adopter l'environnement de *user*

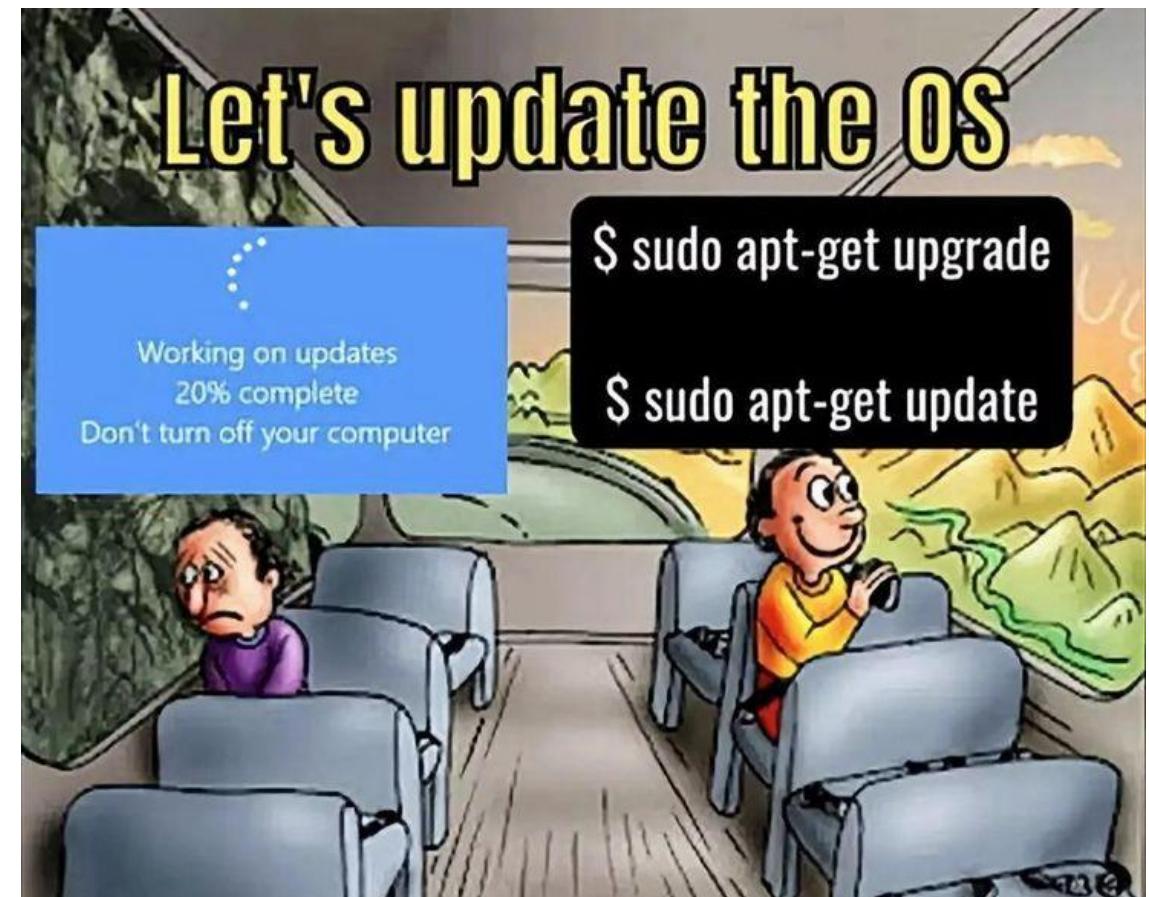
# Installation d'applications sous Linux

- Pour installer une application exécuter la commande suivante en tant que root ou via sudo :
- **Syntaxe** : sudo apt-get install applications
- **Exemple** : Installer virtualbox, lftp, cups-pdf, pdftk, dia, gimp et git  
➤sudo apt-get install virtualbox lftp cups-pdf pdftk dia gimp git

# Maintenir le système à jour

Pour mettre le système à jour, exécuter les deux commandes suivantes en tant que *root* ou via *sudo* :

- sudo apt-get update
- sudo apt-get upgrade



# Chemin absolu, Chemin relatif

**Chemin absolu:** Chemin qui part depuis la racine du système (/ sous Linux, une lettre de lecteur comme C:\ sous Windows).

Exemple :

- Linux : /home/user/documents/fichier.txt
- Windows : C:\Users\Ali\Documents\fichier.txt

**Chemin relatif:** Chemin par rapport au dossier actif (. pour courant, .. pour le dossier parent).

- Exemple :
- Linux : ./fichier.txt ou ../dossier/fichier.txt
- Windows : .\\fichier.txt ou ..\\dossier\\fichier.txt



# Gestion des fichiers : Liens physiques et liens symboliques

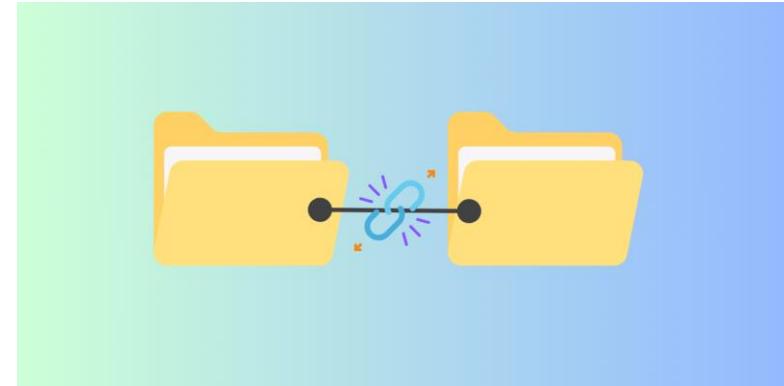
Différence entre lien symbolique et lien physique **sous linux**:

**Lien symbolique** : comme un raccourci, pointe vers un fichier ou dossier

- Syntaxe: `ln -s fichier_original lien_symbolique`
- Exemple: `ln -s /etc/app/config.conf ~/config.conf`

**Lien physique** : copie réelle, partage le même inode

- Syntaxe : `ln fichier_original lien_physique`
- Exemple: `ln notes.txt copie_notes.txt`



**Vérification** : `ls -l` (le lien symbolique affiche une flèche →)

**Suppression** : `rm lien` (ne supprime pas la cible)

# Gestion des fichiers : Liens physiques et liens symboliques

- **Lien physique sous windows**
- Syntaxe : fsutil hardlink create NouveauLien FichierOriginal
- Exemple : fsutil hardlink create C:\data\copie.txt C:\data\original.txt
  
- **Lien symbolique sous windows**
- Syntaxe :mklink [options] Lien Cible
- Exemple vers un fichier: mklink C:\data\lien.txt C:\data\original.txt
- Exemple vers un dossier:  
    mklink /D C:\data\lien\_dossier C:\data\original\_dossier

# Gestion des archives

- L'**archivage** permet de regrouper plusieurs fichiers dans un seul conteneur, tandis que la **compression** réduit leur taille pour gagner de l'espace disque ou accélérer les transferts. Bien souvent, ces deux opérations sont combinées : on archive d'abord avec tar, puis on compresse le tout avec gzip, bzip2 ou xz.
- La commande **tar** (abréviation de *Tape Archive*) est historiquement conçue pour rassembler plusieurs fichiers en une seule archive
- Exemple: tar -cvf archive.tar dossier/



# Gestion des archives

- **Archivage multiple** :archiver plusieurs fichiers ou répertoires en même temps :

Exemple :`tar -cvf sauvegarde.tar fichier1.txt dossier1/ fichier2.log`

- **Archivage relatif vs. absolu**

Par défaut, tar enregistre les **chemins relatifs** des fichiers ajoutés, lors l'extraction de l'archive, les fichiers se replacent par rapport au dossier actuel.

Si vous archivez avec un **chemin absolu**, il est recommandé d'utiliser l'option `--absolute-names` ou `-p` si vous souhaitez **restaurer à l'identique**, ou d'éviter cette option pour ne pas écraser des fichiers critiques en cas d'extraction imprudente.

# Gestion des archives

- **Vérification du contenu d'une archive**

Avant d'extraire ou d'envoyer une archive, il est souvent utile de **contrôler son contenu**. Cela se fait avec :

Exemple : tar -tvf projets.tar

➤ taille, permissions, date, chemin.

- **Extraction d'archives**

Pour extraire une archive dans le répertoire actuel :

Exemple : tar -xvf projets.tar

# Compression et décompression

- **Compression avec gzip**

**gzip** est l'outil de compression le plus courant. Il est rapide et largement compatible. Pour compresser un fichier :

Exemple : gzip fichier.log

- **Décompression avec gunzip**

Pour **décompresser** un fichier .gz :

Exemple: gunzip fichier.log.gz

Ou, avec gzip -d : Exemple : gzip -d fichier.log.gz

➤ De même avec **zip -r**, **bzip2** ou **xz (compression)** et **unzip**, **bunzip2** ou **unxz(decompression)**

# Archivage et compression/ décompression

## **Créer une archive compressée avec gzip (.tar.gz ou .tgz)**

- Exemple : tar -czvf archive.tar.gz dossier/
- -z : compression gzip

## **Décompresser une archive .tar.gz ou .tgz (gzip)**

- Exemple : tar -xzvf archive.tar.gz
- -z : indique que l'archive est compressée avec gzip



# Exercice 1

En se basant sur le fichier inventaire.txt dont le contenu est le suivant:

Nom,PC,IP,OS

Ali,PC01,192.168.1.10,Windows 10

Sara,PC02,192.168.1.11,Ubuntu 22.04

Yassine,PC03,192.168.1.12,Windows 11

Imane,PC04,192.168.1.13,Ubuntu 22.04

Ali,PC01,192.168.1.10,Windows 10

1.Extraire les **noms et systèmes d'exploitation**

2.Trier les lignes extraites

3.Supprimer les doublons

4.Compter le nombre de configurations uniques

# Exercice 2

- Mettre à jour la liste des paquets :
- Installer logiciel htop:
- Créer un fichier dans /etc :
- Créer un **lien physique** vers ce fichier:
- Créer un **lien symbolique** vers ce fichier:
- Vérifier les différences entre les deux types de liens:
- Supprimer ce fichier :

# Exercice 2

- Mettre à jour la liste des paquets :

- Installer logiciel htop:

- Créer un fichier dans /etc :

- Créer un **lien physique** vers ce fichier:

```
sudo ln /etc/config_test.txt /etc/lien_physique.txt
```

- Créer un **lien symbolique** vers ce fichier:

```
sudo ln -s /etc/config_test.txt /etc/lien_symbolique.txt
```

- Vérifier les différences entre les deux types de liens:

```
ls -li /etc/config_test.txt /etc/lien_physique.txt /etc/lien_symbolique.txt
```

- Supprimer ce fichier :  
`sudo rm /etc/test_sudo.txt`

# Exercice 3

Pour chaque chemin suivant identifier si absolu ou relatif

- /home/user/Downloads :
- ../Reports :
- ~/Documents/Reports :
- /var :
- docs :
- / :

# Exercice 3

Pour chaque chemin suivant identifier si absolu ou relatif

- /home/user/Downloads : absolu
- ../Reports : relatif
- ~/Documents/Reports : absolu
- /var : absolu
- docs : relatif
- / : absolu

## Exercice 4:

- Créer un fichier nommé partage.txt dans le répertoire /home/sio avec nano, et y écrire "Ceci est un fichier partagé".
- Utilisez la commande cd pour naviguer vers le répertoire /home/sio.
- Installer et utilisez l'éditeur de texte nano pour créer le fichier.
- Écrivez "Ceci est un fichier partagé" dans le fichier.
- Enregistrez et quittez nano.
- Vérifier que le fichier a bien été créé.
- Rechercher le mot "partagé" dans le fichier et afficher la ligne correspondante
- Utilisez la commande cat partage.txt pour afficher le contenu du fichier "partage.txt".
- Convertir tout le texte en majuscules.
- Afficher le nombre de mots, de lignes et de caractères dans le fichier