

# Chapitre 1

## Ubuntu

### 1.1 WSL

Pour les utilisateurs du système d'exploitation Windows , il est possible d'installer WSL (Windows Subsystem for Linux) , une fonctionnalité qui permet d'exécuter des commandes Linux directement sous Windows, sans avoir à diviser son système en deux (dual-boot) ou à utiliser une machine virtuelle. Cet outil est particulièrement utile pour ceux qui souhaitent bénéficier des avantages des deux systèmes d'exploitation en même temps.

#### 1.1.1 Commandes de base

##### Navigation

- **pwd** : signifie Print Working Directory . Lorsque tu utilises cette commande dans le terminal, elle affiche le chemin complet du répertoire dans lequel tu te trouves actuellement.
- **cd** : signifie change directory, Cette commande permet de changer le répertoire courant dans lequel l'utilisateur travaille.
- **ls** est utilisée pour lister le contenu d'un répertoire.
- **mkdir** : Créer un ou plusieurs répertoires (dossiers).
- **touch** : Créer un fichier vide ou mettre à jour la date de modification d'un fichier existant.
- **rm** : Supprimer des fichiers ou des dossiers.
- **cp** : Copier des fichiers ou des dossiers.
- **mv** : Déplacer ou renommer des fichiers/dossiers.
- **cat** : Afficher, concaténer ou écrire le contenu d'un ou plusieurs fichiers dans le terminal.
- **tree** : Affiche l'arborescence.
- **less** : Permet de visualiser un fichier texte page par page.
- **head** : Affiche les 10 premières lignes d'un fichier.
- **tail** : Affiche les 10 dernières lignes d'un fichier.
- **zip** / **unzip** : Comprime/décomprime.

**Challenge 1** : Crée une structure hiérarchique comme suit :

```

projet/
├── docs/
│   └── README.txt
├── src/
│   └── main.sh
└── data/
    └── input.txt

```

1. Remplis chaque fichier avec un peu de contenu (ex : "Ceci est le README" dans README.txt, etc.).
2. Déplace le fichier input.txt vers le dossier docs. Affiche l'arborescence finale avec tree.
3. Renomme le dossier src en scripts.
4. Supprime le dossier data et tout son contenu.

**Challenge 2 :** Tu es administrateur d'un petit serveur. Ton collègue a laissé des fichiers désorganisés dans son dossier personnel user\_backup. Tu dois :

- Organiser les fichiers.
- Sauvegarder les importants.
- Supprimer les inutiles.
- Compresser les éléments critiques pour archivage ou transfert.

Crée un dossier nommé user\_backup et à l'intérieur, crée les dossiers suivants :

```

user_backup/
├── important/
│   └── temporaire/
│       └── rapport/

```

1. Dans user\_backup, crée les fichiers suivants : todo.txt, note1.txt, brouillon.txt, test1.log, test2.log.
2. Déplace todo.txt et note1.txt vers important.
3. Déplace brouillon.txt, test1.log, et test2.log vers temporaire.
4. Copie important/todo.txt vers rapport sous le nom todo\_backup.txt.
5. Concatène les contenus de todo.txt et note1.txt dans un nouveau fichier synthese.txt dans rapport.
6. Affiche le début de synthese.txt (10 premières lignes).
7. Supprime tous les fichiers dans temporaire qui ont .log.
8. Supprime le dossier temporaire complètement après nettoyage.
9. Affiche le chemin absolu de user\_backup.
10. Liste tout le contenu restant avec arborescence.
11. Comprime le dossier important dans une archive ZIP nommée important\_backup.zip.
12. Crée un dossier restore/ et extrait l'archive important\_backup.zip dedans pour vérifier que la sauvegarde fonctionne.

## Gestion des permissions

- **chmod** : Modifier les droits d'accès d'un fichier ou dossier.

- **chown** : Changer le propriétaire et/ou le groupe d'un fichier ou dossier.
- **sudo** : Permet d'exécuter une commande en tant qu'administrateur
- **su** : Change d'utilisateur (switch user ) dans le terminal.
- **passwd** : Permet de changer le mot de passe d'un utilisateur.
- **adduser et deluser** : Gère les comptes utilisateurs .
- **groups** : Affiche les groupes auxquels appartient un utilisateur.
- **addgroup / delgroup** : Créer ou supprimer un groupe .
- **whoami** : Affiche le nom de l'utilisateur actuel connecté au terminal.

**Challenge 3** : Ce défi est la suite du Challenge 2.

1. Vérifie les permissions actuelles du dossier user\_backup
2. Change les permissions du dossier rapport/ pour qu'il soit accessible uniquement par toi
3. Change les permissions du fichier synthese.txt pour qu'il soit lisible mais non modifiable par les autres utilisateurs
4. Crée un utilisateur au choix (ex : collaborateur)
5. Crée un groupe (ex : equipe-projet) et ajoute l'utilisateur créé dedans
6. Change le propriétaire du dossier important/ vers l'utilisateur créé
7. Affiche les groupes auxquels tu appartiens

## Script Shell

Un script shell est un fichier texte contenant une série de commandes Linux exécutables. Il permet d'automatiser des tâches répétitives, comme :

- Créer des dossiers,
- Nettoyer des fichiers,
- Surveiller des logs,
- Gérer des sauvegardes.

**Challenge 4** : Créer un script Bash qui :

1. Demande le nom de l'utilisateur.
2. Affiche un message de bienvenue personnalisé.
3. Affiche le nom de l'utilisateur actuellement connecté.
4. Donner les autorisations d'exécution au script bash.

**Challenge 5** : Crée un script Bash qui réalise automatiquement les opérations suivantes :

1. Créer un dossier principal nommé mon\_projet, contenant trois sous-dossiers : documents, logs et sauvegardes.
2. Générer un fichier de log nommé logdata.txt dans le dossier logs, contenant les nombres de 1 à 50 (un nombre par ligne).
3. Générer deux fichiers texte : fichierA.txt contenant la ligne "contenu du fichier A" dans mon\_projet/documents/, et fichierB.txt contenant la ligne "Contenu du fichier B" dans mon\_projet/documents/
4. Générer un fichier rapport\_date.txt (ex. rapport\_2025-06-25.txt) dans le dossier mon\_projet, contenant : La date de génération, Le nombre de fichiers présents dans documents et Le nombre de fichiers présents dans logs