

---

## TD 1 : SE

---

### 1 Partie 1 : Rappel des Commandes

#### 1.1 Commandes de Base et d'Orientation

1. `whoami` : Affiche l'utilisateur actuellement connecté.
2. `hostname` : Affiche le nom de la machine.
3. `date` : Affiche la date et l'heure du système.
4. `clear` (ou `Ctrl+L`) : Efface le contenu du terminal.
5. `man <commande>` : Affiche le manuel d'utilisation d'une commande (essentiel !).
6. `history` : Liste les dernières commandes tapées.

#### 1.2 Navigation dans le Système de Fichiers

1. `pwd` (Print Working Directory) : Affiche le chemin du répertoire courant.
2. `ls` (List) : Liste le contenu du répertoire courant.
  - (a) `ls -l` : Affichage long (permissions, propriétaire, taille, date).
  - (b) `ls -a` (All) : Affiche tous les fichiers, y compris les fichiers cachés.
  - (c) `ls -lh` : Affichage long avec des tailles "lisibles" (human-readable).
3. `cd <chemin>` (Change Directory) : Change le répertoire courant.
  - (a) `cd /` : Aller à la racine.
  - (b) `cd ~` (ou `cd`) : Aller au répertoire personnel (Home).
  - (c) `cd ..` : Remonter au répertoire parent.
  - (d) `cd -` : Revenir au répertoire précédent.

#### 1.3 Gestion des Fichiers et Répertoires

1. `touch <fichier>` : Crée un fichier vide.
2. `mkdir <répertoire>` : Crée un nouveau répertoire.
  - (a) `mkdir -p A/B/C` : Crée l'arborescence complète (parents).
3. `cat <fichier>` : Affiche le contenu d'un fichier (petits fichiers).
4. `less <fichier>` : Affiche le contenu d'un fichier page par page (Quitter avec `q`).
5. `head -n 5 <fichier>` : Affiche les 5 premières lignes.
6. `tail -n 5 <fichier>` : Affiche les 5 dernières lignes.
7. `cp <source> <destination>` (Copy) : Copie un fichier ou un répertoire.

- (a) `cp -r Repertoire1/ Repertoire2/` : Copie un répertoire (récursif).
- 8. `mv <source> <destination>` (Move) : Déplace ou renomme.
- 9. `rm <fichier>` (Remove) : Supprime un fichier.
- 10. `rmdir <répertoire>` : Supprime un répertoire (doit être vide).
- 11. `rm -r <répertoire>` (Recursive) : Supprime un répertoire et son contenu (dangereux !).
  - (a) `rm -rf <répertoire>` : Force la suppression récursive sans confirmation.

## 1.4 Recherche de Fichiers

- 1. `find <chemin> -name "<motif>"` : Recherche des fichiers par nom.
  - (a) Ex : `find /home/user -name "*.txt"`
- 2. `grep "<texte>" <fichier>` : Recherche une chaîne de caractères dans un fichier.
  - (a) `grep -i "erreur" logs.txt` : Ignore la casse.
  - (b) `grep -r "config" /etc/` : Recherche récursivement.

## 1.5 Gestion des Processus

- 1. `ps` (Process Status) : Affiche les processus de l'utilisateur.
  - (a) `ps aux` (ou `ps ef`) : Affiche tous les processus du système.
- 2. `top` (ou `htop`) : Affiche les processus en temps réel (Quitter avec `q`).
- 3. `&` (en fin de commande) : Lance la commande en arrière-plan (background).
  - (a) Ex : `sleep 60 &`
- 4. `jobs` : Liste les processus lancés en arrière-plan.
- 5. `fg %<id_job>` (Foreground) : Ramène un processus en avant-plan.
- 6. `bg %<id_job>` (Background) : (Ré)active un processus stoppé en arrière-plan.
- 7. `Ctrl+C` (Signal SIGINT) : Tue (interrompt) le processus en avant-plan.
- 8. `Ctrl+Z` (Signal SIGTSTP) : Stoppe (met en pause) le processus en avant-plan.
- 9. `kill <PID>` : Envoie le signal de terminaison (SIGTERM, 15).
- 10. `kill -9 <PID>` (ou `kill -SIGKILL <PID>`) : Force la terminaison (SIGKILL).
- 11. `pkill <nom>` : Tue un processus par son nom.

## Exercice 1

# 2 Partie 2 : Déroulé du TP (4 heures)

## 2.1 Heure 1 : Exploration et Commandes de Base (60 min)

**Objectif :** Se familiariser avec le terminal et l'arborescence.

### Exercice 2

- (a) Ouvrez un terminal.
- (b) Qui êtes-vous ? (`whoami`)
- (c) Sur quelle machine travaillez-vous ? (`hostname`)
- (d) Où vous trouvez-vous ? (`pwd`)
- (e) Listez le contenu de votre répertoire personnel (`ls -lh`) et analysez la sortie.
- (f) Listez aussi les fichiers cachés (`ls -lha`). Que remarquez-vous ?

### Exercice 3

- (a) Déplacez-vous à la racine du système (`cd /`). Listez le contenu.
- (b) Identifiez les répertoires majeurs : `/bin`, `/etc`, `/home`, `/var`.
- (c) Retournez à votre répertoire personnel (`cd ~`).
- (d) Utilisez `cd ..` pour remonter d'un niveau.
- (e) Revenez au répertoire racine (`/`).
- (f) Revenez d'un coup à votre répertoire précédent (`/home`) en utilisant `cd -`.

### Exercice 4

- (a) Affichez le manuel de la commande `ls` (`man ls`).
- (b) Cherchez comment trier les fichiers par taille (Indice : `-S`).
- (c) Affichez le manuel de `ps` (`man ps`). Cherchez comment afficher l'arborescence des processus (Indice : "forest").
- (d) Utilisez `history` pour voir les 10 dernières commandes tapées (Indice : `history | tail -n 10`).

## 2.2 Heure 2 : Gestion des Fichiers (60 min)

**Objectif :** Créer et manipuler une structure de projet.

### Exercice 5

- (a) Dans votre répertoire `~`, créez un répertoire `Module_OS`.
- (b) Déplacez-vous dans `Module_OS`.
- (c) Créez l'arborescence suivante en une seule commande :

```
Cours/  
  Chapitre1/  
  Chapitre2/  
TP/  
  TP1_Linux/  
  TP2_Processus/
```

(Indice : `mkdir -p ...`)

(d) Vérifiez votre structure (utilisez `ls -R`).

### Exercice 6

(a) Allez dans `TP/TP1_Linux/`.

(b) Créez un fichier vide `commandes.txt` (touch).

(c) Écrivez "Rappel des commandes" dans ce fichier (Indice : `echo "... " > fichier.txt`).

(d) Affichez son contenu (`cat`).

(e) Ajoutez "Fin du rappel" à la suite de ce fichier (Indice : `echo "... " » fichier.txt`).

(f) Créez une copie `commandes_backup.txt` (`cp`).

(g) Visualisez un fichier système long (ex : `/var/log/syslog`) avec `less`.

(h) Affichez les 20 dernières lignes de ce log en temps réel (Indice : `tail -f ...`).

### Exercice 7

(a) Renommez le répertoire `Module_OS/Cours` en `Module_OS/Documentation` (`mv`).

(b) Déplacez `TP/TP1_Linux/commandes_backup.txt` dans `TP/TP2_Processus/`.

(c) Supprimez `TP/TP1_Linux/commandes.txt` (`rm`).

(d) Supprimez le répertoire `TP/TP1_Linux/` (qui doit être vide) avec `rmdir`.

(e) Supprimez \*entièrement\* le répertoire `Documentation` (Indice : `rm -r ...`).

## 2.3 Heure 3 : Recherche et Redirection (60 min)

**Objectif :** Trouver des informations et combiner des commandes.

### Exercice 8

i. Recherchez le mot "user" dans le fichier `/etc/passwd`.

ii. Recherchez "USER" en ignorant la casse (`grep -i`).

iii. Comptez le nombre de lignes contenant "user" (`grep -c`).

iv. Recherchez récursivement "network" dans `/etc/` (Indice : `grep -r "network" /etc/ 2> /dev/null`).

### Exercice 9

i. Depuis `~`, trouvez tous les fichiers finissant par `.log` (`find . -name "*.log"`).

ii. Trouvez tous les répertoires (type `d`) nommés `TP1_Linux`.

iii. Trouvez tous les fichiers (type `f`) de plus de 1Mo (`-size +1M`) dans `/var`.

### Exercice 10

i. Listez les fichiers dans `/bin` et filtrez pour ne voir que ceux contenant "z" (`ls /bin | grep "z"`).

ii. Affichez tous les processus (`ps aux`) et filtrez pour ne voir que ceux de root (`... | grep "root"`).

iii. Affichez votre historique et filtrez les commandes `cd` (`history | grep "cd"`).

## 2.4 Heure 4 : Gestion des Processus (60 min)

**Objectif :** Comprendre le cycle de vie d'un processus.

### (a) Exercice 4.1 : Monitoring (15 min)

- i. Lancez `top`. Identifiez PID, %CPU, %MEM, USER.
- ii. Triez par Mémoire (Touche M) puis CPU (Touche P). Quittez (q).
- iii. Affichez tous vos processus (`ps`).
- iv. Affichez tous les processus système (`ps aux`). Repérez le processus `init` ou `systemd`. Quel est son PID ?

### Exercice 11

- i. Lancez `sleep 300`. Le terminal est bloqué. Arrêtez avec `Ctrl+C`.
- ii. Lancez `sleep 300 &`. Vérifiez son statut avec `jobs`. Notez son PID.
- iii. Lancez `ping localhost > /dev/null &`.
- iv. Listez vos "jobs" (`jobs`).
- v. Ramenez le `sleep` en avant-plan (`fg %1`).
- vi. Stoppez-le (mettez en pause) avec `Ctrl+Z`.
- vii. Vérifiez l'état avec `jobs` (doit être "Stopped").
- viii. Ré-activez-le en arrière-plan (`bg %1`).

### Exercice 12

- i. Ouvrez un \*second terminal\*.
- ii. Dans le T2, trouvez le PID du `sleep 300` (`ps aux | grep "sleep"`).
- iii. Tuez-le "proprement" (`kill <PID_sleep>`). Vérifiez dans le T1 avec `jobs`.
- iv. Dans le T1, lancez un processus infini : `yes > /dev/null &`.
- v. Observez sa consommation CPU avec `top`.
- vi. Tuez-le par son nom (`pkill yes`).
- vii. Relancez-le. Tentez de le tuer avec le signal SIGKILL (`pkill -9 yes`).

**Discussion :** Quelle est la différence entre `kill` (SIGTERM) et `kill -9` (SIGKILL) ?