

Linux : TP1

Halim Djerroud

révision 1.0

1 Préambule : Premier contact avec le terminal

1.1 Lancement du terminal

1. Ouvrir un terminal sur votre distribution (Debian/Ubuntu/CentOS/Kali) :
 - **Raccourci clavier** : Ctrl + Alt + T
 - **Interface graphique** : Menu Applications → Terminal
2. Observez l'invite de commande (prompt) :
 - `utilisateur@machine:~$` pour un utilisateur normal
 - `root@machine:~#` pour l'administrateur

Attention :

En environnement GNU/Linux, il est recommandé d'effectuer toutes les tâches quotidiennes avec un compte utilisateur normal. Le compte administrateur (root) doit être utilisé uniquement pour les opérations de maintenance ou de configuration système

3. Tapez votre première commande :

```
whoami
```

Elle affiche le nom de l'utilisateur courant.

4. Testez également :

```
id  
hostname  
date
```

Rappel :

Une commande Linux suit généralement cette structure :

`commande [options] [arguments]`

Exemple : `ls -la /home` où :

- `ls` est la commande
- `-la` sont les options
- `/home` est l'argument

2 Exercice 1 : Où suis-je, qui suis-je ?

1. Ouvrez un terminal.
2. Quel est le nom de votre machine ?
3. Quel est votre nom d'utilisateur ?
4. Quels sont vos privilèges ?
5. Donnez le chemin absolu de votre répertoire par défaut.

3 Navigation dans l'arborescence

3.1 Exercice 1

1. Placez-vous dans votre répertoire personnel et affichez son chemin complet :

```
cd ~
pwd
```

2. Créez la structure suivante dans votre répertoire `tp_arborescence` :

```
tp_arborescence/
|- docs/
|- src/
|- images/
|   |- icones/
|- scripts/
```

(Astuce : vous pouvez utiliser `mkdir -p` pour créer l'ensemble des répertoires en une seule commande.)

3. Déplacez-vous successivement :
 - dans le dossier `docs`
 - puis dans `images/icones` (utiliser le chemin relatif)
 - revenez ensuite au répertoire `tp_arborescence`
4. Utilisez les commandes suivantes et expliquez le résultat :

```
cd ..
cd -      #old dir
cd .
```

5. Supprimez le dossier `tp_arborescence` et tout son contenu en une seule commande.

3.2 Exercice 2 : Manipulation de fichiers et répertoires

1. Dans votre répertoire personnel, créez un dossier `tp_fichiers` et placez-vous dedans.
2. Créez trois fichiers vides : `note1.txt`, `note2.txt`, `rapport.txt`.
3. Vérifiez le contenu du répertoire avec différentes options de `ls` :
4. Copiez le fichier `rapport.txt` dans un nouveau fichier `rapport_copie.txt`.
5. Déplacez le fichier `note2.txt` dans un sous-dossier `archives` (à créer).
6. Supprimez le fichier `note1.txt`, puis supprimez le dossier `archives` et son contenu.

4 Exercice 3 : Utiliser l'aide sous Linux

1. Affichez le manuel complet de la commande `ls` :

```
man ls
```

Quelle option permet d'afficher les fichiers cachés ?

2. Recherchez toutes les pages de manuel contenant le mot-clé `process` (Astuce : Lisez `man man`)
3. Utilisez la documentation `info` pour la commande `ls` :

```
info ls
```

Naviguez avec `Espace` (page suivante), `b` (page précédente), `q` (quitter).

4. Affichez l'aide rapide intégrée :

```
ls -h
ls --help
```

Quelle différence constatez-vous entre `-h` et `--help` ?

5. Comparez les résultats de `whatis` et `apropos` :

```
whatis ls
apropos ls
```

Quelle est la différence d'usage entre ces deux commandes ?

5 Exercice 4 : Lecture et analyse de fichiers

1. Affichez le contenu du fichier `/etc/passwd` avec la commande :

```
cat
```

2. Parcourez le même fichier page par page avec la commande `less` (Astuce : utilisez `/motif` pour rechercher un mot, `q` pour quitter.)
3. Affichez uniquement les 5 premières lignes du fichier :

```
/etc/passwd
```

4. Affichez uniquement les 5 dernières lignes du fichier :

```
/etc/passwd
```

5. Surveillez en temps réel les nouvelles écritures dans un fichier de log :

```
/var/log/syslog
```

6. Identifiez le type d'un fichier (texte, binaire, exécutable, image, etc.) :

```
/etc/passwd  
/bin/ls
```

6 Exercice 5 : Édition de fichiers avec nano

1. Créez et ouvrez un fichier texte nommé `notes.txt` avec `nano` :

```
nano notes.txt
```

2. Dans l'éditeur, saisissez quelques lignes de texte (par exemple une liste de courses ou des notes de TP).
3. Sauvegardez votre fichier et quittez l'éditeur. (Astuce : `Ctrl + O` pour enregistrer, puis `Ctrl + X` pour quitter.)
4. Ré-ouvrez le fichier `notes.txt` pour vérifier que vos modifications ont bien été prises en compte.
5. Ajoutez une nouvelle ligne à la fin du fichier, sauvegardez et quittez de nouveau.
6. Affichez le contenu du fichier directement dans le terminal.

7 Exercice 6 : Rechercher avec grep

1. Affichez toutes les lignes du fichier `/etc/passwd` contenant le mot `bash`.
2. Affichez toutes les lignes du fichier `/etc/passwd` qui **ne contiennent pas** `nologin`.
3. Cherchez de manière insensible à la casse le mot `root` dans le fichier `/etc/passwd`.
4. Recherchez récursivement dans le répertoire `/etc` les fichiers contenant le mot `network`.

8 Mini-projet : Analyse d'un répertoire système (à terminer avant la prochaine séance)

Vous allez créer une arborescence de travail, y placer des fichiers, puis utiliser les différentes commandes étudiées pour analyser et manipuler ces fichiers.

1. Créez une arborescence de projet dans votre répertoire personnel :

```
projet_linux/  
|-- docs/  
|-- data/  
|   |-- logs/  
|-- scripts/
```

2. Dans le dossier `docs`, créez avec `nano` un fichier `readme.txt` contenant une brève description du projet (2 ou 3 lignes).
3. Dans le dossier `data/logs`, copiez le fichier `/etc/passwd` sous le nom `users.txt`.

4. Utilisez les commandes suivantes pour explorer et analyser ce fichier :
 - **file** pour déterminer son type.
 - **head** et **tail** pour consulter quelques lignes.
 - **less** pour parcourir le fichier page par page.
 - **cat** pour afficher son contenu complet.
5. Recherchez avec **grep** :
 - Les lignes contenant **bash**.
 - Le nombre d'utilisateurs dont le shell est **nologin**.
 - Les lignes contenant **root** (insensible à la casse).
6. Créez un script ¹ **analyse.sh** dans le dossier **scripts** qui affiche :
 - La date et l'utilisateur courant.
 - Le nombre total de lignes dans **users.txt**.
 - Le nombre d'utilisateurs ayant **bash** comme shell.
 - Les 5 premiers et les 5 derniers utilisateurs du fichier.
7. Rendez le script exécutable et lancez-le.

1. Un script est un fichier texte contenant une suite de commandes exécutées automatiquement par un interpréteur (par ex. **bash**). Un script permet d'automatiser des tâches répétitives en regroupant plusieurs commandes dans un seul fichier exécutable