

Gestion des Fichiers sous Linux

Introduction : Tout est Fichier sous Linux

- **Principe Fondamental** : Sous Linux, tout est traité comme un fichier :
 - Fichiers classiques (texte, image, vidéo).
 - Répertoires (fichiers spéciaux contenant d'autres fichiers).
 - Périphériques, connexions réseau, processus.
 - **Structure Arborescente** :
 - Racine / : point de départ de l'arborescence.
 - **Répertoires** : nœuds.
 - **Fichiers** : feuilles de l'arborescence.
-

Chemins dans le Système de Fichiers

1. **Chemin Absolu** :
 - Part toujours de la racine /.
 - Syntaxe : chemin complet jusqu'au fichier (ex. : `/home/user/document.txt`).
 2. **Chemin Relatif** :
 - Basé sur le répertoire courant.
 - Utilisation de `..` pour remonter dans l'arborescence (ex. : `../document.txt`).
 3. **Dossier Personnel (~)** :
 - `~` réfère au répertoire personnel de l'utilisateur.
 - Exemple : `cd ~` pour accéder à `/home/user`.
-

Commandes Essentielles pour les Fichiers

1. Afficher le Répertoire Courant

- **Commande** : `pwd` (print working directory).
- **But** : Montrer le chemin absolu du répertoire actuel.
- **Exemple** :

```
pwd
```

```
Sortie : /home/user.
```

2. Lister les Fichiers

- **Commande** : `ls [options] [répertoires]`.
- **Options Principales** :
 - `-l` : Format long (détails).
 - `-a` : Inclut les fichiers cachés.

- **-R** : Liste récursive.
- **Exemple :**

```
ls -l /home
```

3. Changer de Répertoire

- **Commande** : `cd [chemin]`.
- **But** : Naviguer entre répertoires.
- **Exemple :**

```
cd /etc
```

4. Créer un Répertoire

- **Commande** : `mkdir [options] répertoires`.
- **Option** : `-p` pour créer une hiérarchie.
- **Exemple :**

```
mkdir -p dossier1/dossier2
```

5. Supprimer un Répertoire

- **Commande** : `rmdir [options] répertoires`.
- **Limite** : Ne supprime que des répertoires vides.
- **Exemple :**

```
rmdir dossier_vide
```

6. Créer un Fichier Vide

- **Commande** : `touch [options] fichier`.
- **But** : Créer un fichier ou mettre à jour ses timestamps.
- **Exemple :**

```
touch fichier.txt
```

7. Éditer un Fichier

- **Commandes :**
 - Graphique : `gedit fichier`.
 - Terminal : `nano fichier`.
- **Exemple :**

```
gedit fichier.txt
```

8. Afficher le Contenu d'un Fichier

- **Commande** : `cat [options] fichier`.
- **Option** : `-n` pour numéroté les lignes.
- **Exemple :**

```
cat -n fichier.txt
```

9. Supprimer un Fichier

- **Commande** : `rm [options] fichiers.`
- **Options** :
 - `-r` : Suppression récursive (y compris des répertoires).
 - `-i` : Confirmation pour chaque fichier.
- **Exemple** :

```
rm -i fichier.txt
```

10. Copier un Fichier

- **Commande** : `cp [options] source destination.`
- **Exemple** :

```
cp fichier.txt copie_fichier.txt
```

11. Déplacer ou Renommer un Fichier

- **Commande** : `mv [options] source destination.`
- **Exemple** :

```
mv fichier.txt nouveau_nom.txt
```

Liens (Physiques et Symboliques)

1. Lien Physique

- **Commande** : `ln fichier lien.`
- **Principe** : Plusieurs noms pour une même donnée sur le disque.
- **Exemple** :

```
ln fichier.txt lien_physique.txt
```

2. Lien Symbolique

- **Commande** : `ln -s fichier lien.`
- **Principe** : Crée un pointeur vers un fichier (même sur une autre partition).
- **Exemple** :

```
ln -s fichier.txt lien_symbolique.txt
```

- **Particularités** :
 - Si le fichier original est supprimé, le lien devient cassé.
 - La taille d'un lien symbolique correspond au chemin qu'il pointe.
-

Exemples Pratiques Complémentaires

- **Lister récursivement tous les fichiers cachés :**

```
ls -aR
```

- **Copier un répertoire avec tous ses fichiers :**

```
cp -r source destination
```

- **Supprimer un répertoire et son contenu sans confirmation :**

```
rm -rf repertoire
```

- **Afficher le chemin absolu d'un fichier spécifique :**

```
readlink -f fichier.txt
```
