

Les pipes (ou tubes) sont une fonctionnalité puissante de Linux qui permet de relier plusieurs commandes ensemble, en utilisant la sortie d'une commande comme entrée pour une autre. Voici une série d'exemples illustrant différentes utilisations des pipes sous Linux :

## Introduction aux Pipes

Le symbole `|` est utilisé pour créer un pipe entre deux commandes.

## Exemples de Base

### 1. Afficher la liste des fichiers et la paginer avec `less` :

```
ls -l | less
```

### 2. Compter le nombre de fichiers dans un répertoire :

```
ls -l | wc -l
```

### 3. Afficher les processus et les filtrer pour trouver un processus spécifique :

```
ps aux | grep "nom_du_processus"
```

### 4. Afficher les 10 lignes les plus fréquentes dans un fichier :

```
cat fichier.txt | sort | uniq -c | sort -nr | head -10
```

### 5. Afficher uniquement les interfaces réseau actives :

```
ifconfig | grep "inet "
```

## Combinaisons Pratiques

### 6. Trouver et lister tous les fichiers `.log` modifiés dans les 7 derniers jours :

```
find /var/log -name "*.log" -mtime -7 | xargs ls -l
```

### 7. Afficher les 10 fichiers les plus grands dans un répertoire :

```
du -ah /chemin/vers/repertoire | sort -rh | head -10
```

**8. Surveiller les logs du système et rechercher les erreurs en temps réel :**

```
tail -f /var/log/syslog | grep "error"
```

**9. Afficher l'utilisation de la mémoire et la paginer avec less :**

```
free -m | less
```

**10. Trouver les fichiers contenant une chaîne de caractères spécifique et afficher les lignes correspondantes :**

```
grep -r "chaîne_de_caracteres" /chemin/vers/repertoire | less
```

Traitement Avancé

**11. Extraire les adresses IP d'un fichier log et afficher les occurrences uniques :**

```
grep -oE '\b([0-9]{1,3}\.){3}[0-9]{1,3}\b' fichier.log | sort | uniq
```

**12. Surveiller l'activité réseau et extraire les connexions vers un port spécifique :**

```
netstat -an | grep ":80 " | grep "ESTABLISHED"
```

**13. Afficher l'historique des commandes et compter les occurrences de chaque commande :**

```
history | awk '{print $2}' | sort | uniq -c | sort -nr | head -10
```

**14. Afficher les 10 fichiers les plus récents modifiés dans un répertoire :**

```
ls -lt /chemin/vers/repertoire | head -10
```

Combinaisons avec des Commandes de Texte

**15. Afficher les lignes contenant une chaîne spécifique et leurs numéros de ligne :**

```
grep -n "chaîne_de_caracteres" fichier.txt | less
```

**16. Compter le nombre de mots dans un fichier texte :**

```
cat fichier.txt | wc -w
```

**17. Convertir tout le texte d'un fichier en majuscules :**

```
cat fichier.txt | tr 'a-z' 'A-Z'
```

**18. Remplacer un mot par un autre dans un fichier et afficher le résultat :**

```
cat fichier.txt | sed 's/ancien_mot/nouveau_mot/g'
```

**19. Extraire les 100 premières lignes d'un fichier, les trier et les afficher :**

```
head -100 fichier.txt | sort | less
```

Utilisation de **tee** pour Diviser la Sortie

**20. Afficher la sortie d'une commande tout en l'enregistrant dans un fichier :**

```
ls -l | tee fichier.txt
```

**21. Exécuter un script et enregistrer la sortie et les erreurs dans des fichiers distincts :**

```
./script.sh 2>&1 | tee sortie.log
```