Commandes et Config

Les commandes Linux constituent l'essence même de l'interaction avec le système d'exploitation à travers un terminal ou une interface en ligne de commande (CLI). Elles permettent d'exécuter une multitude de tâches, allant de la navigation dans le système de fichiers à la gestion des processus, en passant par la configuration réseau et la gestion des utilisateurs.

Le système Linux repose fortement sur la philosophie UNIX, où chaque commande effectue une tâche spécifique. Ces commandes peuvent être combinées en utilisant des pipes et des redirections pour créer des scripts puissants et automatiser des processus complexes. Les utilisateurs de Linux apprécient la polyvalence et la puissance des commandes, qui offrent un contrôle total sur le système.

Qu'il s'agisse de simples commandes comme ls pour lister les fichiers ou de commandes plus complexes comme grep pour rechercher du texte à travers plusieurs fichiers, maîtriser les commandes Linux est une compétence précieuse pour les administrateurs système, les développeurs et les utilisateurs expérimentés.

Commandes Basique

Quelques commandes essentielles pour débuter :

- **`ls`** : Affiche la liste des fichiers et répertoires dans un répertoire donné. Exemples :
  - `ls` : Liste les fichiers dans le répertoire courant.
  - `ls -l` : Liste les fichiers avec des détails comme les permissions, la taille et la date de modification.
- **`cd`** : Change le répertoire courant. Exemples :
  - `cd /home/user/Documents` : Se déplace vers le répertoire `Documents`.
  - `cd ..` : Revient au répertoire parent.
- **`cp`** **et** **`mv`** : Copie ou déplace des fichiers/répertoires.
  - `cp file1.txt /home/user/` : Copie `file1.txt` vers le répertoire `/home/user/`.
  - `mv file1.txt /home/user/` : Déplace `file1.txt` vers le répertoire `/home/user/`.
- **`rm`** : Supprime des fichiers ou des répertoires. Attention à l'utilisation car cette commande ne place pas les fichiers dans une corbeille.
  - `rm file1.txt` : Supprime le fichier `file1.txt`.
  - `rm -r /home/user/dir` : Supprime récursivement le répertoire `dir` et son contenu.
- **`chmod`** **et** **`chown`** : Modifie les permissions et le propriétaire d'un fichier ou d'un répertoire.
  - `chmod 755 script.sh` : Donne les permissions de lecture, écriture et exécution au propriétaire, et les permissions de lecture et exécution aux autres.
  - `chown user:user file1.txt` : Change le propriétaire et le groupe de `file1.txt`.
- **`ps`** **et** **`top`** : Affiche les processus en cours.
  - `ps aux` : Affiche tous les processus en cours avec des informations détaillées.
  - `top` : Affiche les processus actifs avec une mise à jour en temps réel.
- **`grep`** : Recherche des chaînes de caractères dans des fichiers.
  - `grep 'mot' fichier.txt` : Recherche toutes les occurrences de `mot` dans `fichier.txt`.
  - `grep -r 'mot' /etc/` : Recherche récursivement `mot` dans tous les fichiers du répertoire `/etc/`.
- **`find`** : Recherche des fichiers dans un répertoire donné.
  - `find /home/user/ -name "*.txt"` : Recherche tous les fichiers `.txt` dans le répertoire `/home/user/`.
- **`tar`** : Archive et compresse des fichiers.
  - `tar -cvf archive.tar /home/user/dir` : Crée une archive `tar` de `dir`.
  - `tar -xvf archive.tar` : Décompresse l'archive `tar`.
- **`ssh`** : Se connecte à un serveur distant en utilisant le protocole SSH.
  - `ssh user@ip_serveur` : Se connecte en SSH au serveur avec l'adresse `ip_serveur`.

- **sudo** : Exécute des commandes avec des privilèges administratifs.
  - sudo apt update : Exécute la commande de mise à jour des paquets avec les droits d'administration.

Configuration de base pour la gestion des commandes :

1. **Bashrc et Aliases :**
   - **~/.bashrc** : Le fichier de configuration des commandes du terminal Bash où tu peux définir des alias pour simplifier les commandes répétitives. Par exemple :
     bash
     ```
     alias ll='ls -alF'
     alias grep='grep --color=auto'
     ```
     Ces alias te permettent de taper ll au lieu de ls -alF pour voir les fichiers détaillés, ou de colorer automatiquement les résultats de grep.
2. **Gestionnaire de paquets :**
   - Selon ta distribution, tu utiliseras des commandes comme apt (Debian/Ubuntu) ou pacman (Arch) pour gérer l'installation, la mise à jour et la suppression de logiciels. Par exemple :
     - Sur Ubuntu : sudo apt install package_name
     - Sur Arch : sudo pacman -S package_name
3. **Scripting Bash :**
   - L'écriture de scripts Bash permet d'automatiser des tâches courantes. Par exemple, un script simple pour sauvegarder un répertoire pourrait ressembler à ceci :
     bash
     ```
     #!/bin/bash
     tar -czvf backup.tar.gz /home/user/dossier
     echo "Sauvegarde terminée."
     ```