

La plupart des gens pensent que Linux est un système d'exploitation compliqué qui n'est utilisé que par les **programmeurs**. Mais apprendre linux n'est pas aussi effrayant qu'il n'y paraît. Nul besoin de prendre un [cours de programmation](#) ou maîtriser un langage de programmation. Dans ce tuto court mais très complet vous allez découvrir **les commandes Linux de base** et comment les utiliser comme un pro.

Linux est une famille entière de systèmes d'exploitation Unix **open-source (à code source ouvert)**, qui sont basés sur le noyau Linux.

Cela inclut tous les systèmes les plus populaires basés sur Linux comme Ubuntu, Fedora, Mint, Debian, et d'autres. Plus précisément, on les appelle des distributions ou des distros.

Depuis sa première publication en **1991**, Linux n'a cessé de gagner en popularité en raison de sa nature open-source. Les gens peuvent librement le modifier et le redistribuer sous leur propre nom.

Lorsque vous utilisez un système d'exploitation Linux, vous devez utiliser un shell – une interface qui vous donne accès aux services du système. La plupart des distributions Linux utilisent une interface utilisateur graphique (GUI) comme shell, principalement pour faciliter l'utilisation par les utilisateurs.

Cela étant dit, il est recommandé d'utiliser une **interface en ligne de commande (CLI)** car elle est plus puissante et plus efficace. En tapant des commandes texte, vous pourrez effectuer les tâches qui nécessitent un processus en plusieurs étapes via l'interface graphique, en quelques secondes seulement.

Donc si vous envisagez d'utiliser le système d'exploitation Linux, l'apprentissage des **commandes de base** sera

très utile. Dans cet article, vous apprendrez **40 commandes Linux de base** qui vous aideront sans aucun doute à naviguer dans Linux en tant que débutant.

Sommaire

- [Qu'est-ce qu'une commande Linux ?](#)
- [Les 40 commandes Linux les plus utilisées](#)
- [Aide-mémoire commandes linux](#)
- [FAQ sur les commandes Linux](#)

Qu'est-ce qu'une

commande Linux ?

Une commande Linux est un programme ou un utilitaire qui s'exécute sur la ligne de commande (CLI) – une console qui interagit avec le système par le biais de textes et de processus. Elle est similaire à l'application Invite de commande de Windows.

Les commandes Linux sont exécutées sur le terminal en appuyant sur Entrée à la fin de la ligne. Vous pouvez exécuter des commandes pour effectuer diverses tâches, de l'installation de paquets à la gestion des utilisateurs en passant par la manipulation de fichiers.

Voici à quoi ressemble la syntaxe générale d'une commande Linux :

Nom commande [option(s)] [paramètre(s)].

Une commande peut contenir une **option** ou un **paramètre**. Dans certains cas, elle peut s'exécuter sans eux. Ce sont les trois parties les plus courantes d'une commande :

- **Nom commande** : c'est la règle que vous souhaitez exécuter.
- **Option ou le drapeau** : modifie le fonctionnement de la commande. Pour l'invoquer, utilisez des tirets (-) ou des doubles tirets (--).
- **Paramètre ou argument** : spécifie toute information nécessaire à la commande.

N'oubliez pas que toutes les commandes Linux sont sensibles à la casse.

Les 40 commandes Linux les plus utilisées

Avant de passer à la liste des commandes de base, vous devez d'abord ouvrir Terminal. Si vous n'êtes toujours pas sûr de vous, consultez notre [tutoriel sur la ligne de commande](#).

Bien que les étapes puissent différer selon votre distribution Linux, l'application Terminal se trouve généralement dans la section **Utilitaires**.

Voici la liste des commandes de base de Linux :

1. Commande sudo

Abréviation de superuser do, **sudo** est l'une des commandes de base les plus populaires de Linux, qui vous permet d'effectuer des tâches nécessitant des autorisations administratives ou de **super utilisateur**. Lors de l'utilisation de sudo, le système demande aux utilisateurs de s'authentifier avec un mot de passe. Ensuite, le système Linux enregistre un horodatage en tant que traceur. Par défaut, chaque utilisateur root peut exécuter des commandes sudo pendant **15 minutes/session**.

Si vous essayez d'exécuter sudo dans la ligne de commande sans vous authentifier, le système enregistrera l'activité en tant qu'événement de sécurité.

Voici la syntaxe générale :

sudo (commande)

Vous pouvez également ajouter une option, par exemple :

- **-k** ou **-reset-timestamp** invalide le fichier d'horodatage.
- **-g** ou **-group=groupe** exécute les commandes en tant que nom de groupe ou ID spécifié.
- **-h** ou **-host=hôte** exécute des commandes sur l'hôte.

2. Commande pwd

Utilisez la commande **pwd** pour trouver le chemin de votre répertoire de travail actuel. En entrant

simplement **pwd**, vous obtiendrez le chemin d'accès complet, c'est-à-dire le chemin de tous les répertoires commençant par une barre oblique (/). Par exemple, **/home/nom d'utilisateur**.

La commande **pwd** utilise la syntaxe suivante :

pwd [option]

Il dispose de 2 options possibles :

- **-L** ou **-logique** affiche le contenu des variables d'environnement, y compris les liens symboliques.
- **-P** ou **-physical** affiche le chemin réel du répertoire actuel.

3. Commande **cd**

Pour naviguer dans les fichiers et les répertoires de Linux, utilisez la commande **cd**. En fonction de votre répertoire de travail actuel, cette commande requiert soit le chemin d'accès complet, soit le nom du répertoire.

L'exécution de cette commande sans option vous conduira au dossier personnel. Gardez à l'esprit que seuls les utilisateurs disposant des privilèges **sudo** peuvent l'exécuter.

Supposons que vous vous trouviez dans le **répertoire /home/nom d'utilisateur/Documents** et que vous souhaitiez accéder à **Photos**, un sous-répertoire de **Documents**. Pour ce faire, entrez la commande suivante :

cd Photos.

Si vous souhaitez passer à un tout nouveau répertoire, par exemple, **/home/nom d'utilisateur/ films**, vous devez saisir **cd** suivi du chemin absolu du répertoire :

cd /home/nom d'utilisateur/ films

Voici quelques raccourcis pour vous aider à naviguer :

- **cd ~[nom d'utilisateur]** permet d'accéder au répertoire personnel d'un autre utilisateur.
- **cd ..** monte d'un répertoire.
- **cd-** permet de revenir au répertoire précédent.

4. Commande ls

La commande **ls** répertorie les fichiers et les répertoires d'un système. Si elle est exécutée sans drapeau ni paramètre, cette commande affiche le contenu du répertoire de travail actuel.

Pour afficher le contenu d'autres répertoires, tapez **ls** suivi du chemin d'accès souhaité. Par exemple, pour afficher les fichiers du dossier **Documents**, entrez :

ls /home/nom d'utilisateur/Documents

Voici quelques options que vous pouvez utiliser avec la commande **ls** :

- **ls -R** dresse la liste de tous les fichiers contenus dans les sous-répertoires.
- **ls -a** affiche les fichiers cachés en plus des fichiers visibles.
- **ls -lh** affiche la taille des fichiers dans des formats facilement lisibles, tels que MB, GB et TB.

5. Commande cat

Concatenate, ou **cat**, est l'une des commandes Linux les plus fréquemment utilisées. Elle énumère, combine et écrit le contenu des fichiers sur la sortie standard. Pour exécuter la commande **cat**, tapez **cat** suivi du nom du fichier et de son extension. Par exemple :

cat filename.txt.

Voici d'autres façons d'utiliser la **commande cat** :

- **cat > nomfichier.txt** crée un nouveau fichier.
- **cat nomfichier1.txt nomfichier2.txt > nomfichier3.txt** fusionne **nomfichier1.txt** et **nomfichier2.txt** et stocke le résultat dans **nomfichier3.txt**.
- **tac nomfichier.txt** affiche le contenu dans l'ordre inverse.

6. Commande cp

La commande **cp** permet de copier des fichiers ou des répertoires et leur contenu. Examinez les cas d'utilisation suivants.

Pour copier un fichier du répertoire actuel vers un autre, entrez **cp** suivi du nom du fichier et du répertoire de destination. Par exemple :

cp nomfichier.txt /home/nomd'utilisateur/Documents

Pour copier des fichiers dans un répertoire, entrez les noms des fichiers suivis du répertoire de destination :

cp nomfichier1.txt nomfichier2.txt nomfichier3.txt /home/nomd'utilisateur/Documents

Pour copier le contenu d'un fichier dans un nouveau fichier du même répertoire, entrez **cp** suivi du fichier source et du fichier de destination :

cp nomfichier1.txt nomfichier2.txt

Pour copier un répertoire entier, passez l'option **-R** avant de taper le répertoire source, suivi du répertoire de destination :

cp -R /home/username/Documents /home/username/Documents_backup

7. Commande mv

La commande mv sert principalement à déplacer et à renommer des fichiers et des répertoires. En outre, elle ne produit aucun résultat à l'exécution.

Il suffit de taper **mv** suivi du nom du fichier et du répertoire de destination. Par exemple, vous souhaitez déplacer **nomfichier.txt** dans le répertoire **/home/nomd'utilisateur/Documents** :

```
mv nom_de_fichier.txt  
/home/nom_d'utilisateur/Documents.
```

Vous pouvez également utiliser la commande **mv** pour renommer un fichier :

```
mv ancien_nom_de_fichier.txt  
nouveau_nom_de_fichier.txt
```

8. Commande mkdir

La commande **mkdir** permet de créer un ou plusieurs répertoires en une seule fois et de définir les autorisations pour chacun d'entre eux. L'utilisateur qui exécute cette commande doit avoir le droit de créer un nouveau dossier dans le répertoire parent, sinon il risque de recevoir un message d'erreur « permission denied » (autorisation refusée).

Voici la syntaxe de base :

```
mkdir [option] nom_du_répertoire
```

Par exemple, vous souhaitez créer un répertoire appelé **Musique** :

```
mkdir Musique
```

Pour créer un nouveau répertoire appelé **Chansons** à l'intérieur de **Musique**, utilisez la commande suivante :

```
mkdir Musique/Chansons
```

La commande **mkdir** accepte de nombreuses options, telles que :

- **-p** ou **-parents** créent un répertoire entre deux dossiers existants. Par exemple, **mkdir -p Music/2020/Songs** créera le nouveau répertoire « 2020 ».
- **-m** définit les droits d'accès aux fichiers. Par exemple, pour créer un répertoire avec des autorisations complètes de lecture, d'écriture et d'exécution pour tous les utilisateurs, entrez **mkdir -m777 nom_du_répertoire**.
- **-v** affiche un message pour chaque répertoire créé.

9. Commande rmdir

Pour supprimer définitivement un répertoire vide, utilisez la **commande rmdir**. N'oubliez pas que l'utilisateur qui exécute cette commande doit disposer des privilèges **sudo** dans le répertoire parent.

Par exemple, vous souhaitez supprimer un sous-répertoire vide nommé **personal1** et son dossier principal **mydir** :

```
rmdir -p mydir/personal1
```

10. Commande rm

La commande **rm** est utilisée pour supprimer des fichiers dans un répertoire. Assurez-vous que l'utilisateur qui exécute cette commande dispose des droits d'écriture. N'oubliez pas l'emplacement du répertoire, car cette opération supprimera le(s) fichier(s) et vous ne pourrez pas l'annuler.

Voici la syntaxe générale :

```
rm nom de fichier
```

Pour supprimer plusieurs fichiers, entrez la commande suivante :

rm nomfichier1 nomfichier2 nomfichier3

Voici quelques options acceptables que vous pouvez ajouter :

- **-i** demande au système de confirmer la suppression d'un fichier.
- **-f** permet au système de procéder à la suppression sans confirmation.
- **-r** supprime les fichiers et les répertoires de manière réursive.

11. Commande touch

La **commande touch** permet de créer un fichier vide ou de générer et de modifier un horodatage dans la ligne de commande.

Par exemple, entrez la commande suivante pour créer un fichier HTML nommé **web** dans le répertoire **Documents** :

touch /home/nom d'utilisateur/Documents/web.html

12. Commande locate

La **commande locate** permet de trouver un fichier dans le système de base de données.

De plus, l'ajout de l'argument **-i** désactive la sensibilité à la casse, ce qui vous permet de rechercher un fichier même si vous ne vous souvenez pas de son nom exact.

Pour rechercher un contenu contenant deux mots ou plus, utilisez un astérisque (*). Par exemple, l'astérisque (*) est utilisé pour rechercher un contenu qui contient deux mots ou plus :

locate -i école*note

La commande recherche les fichiers contenant les mots **école** et **note**, qu'ils soient en majuscules ou en minuscules.

13. Commande find

La commande **find** permet de rechercher des fichiers dans un répertoire spécifique et d'effectuer les opérations suivantes. Voici la syntaxe générale :

find [option] [chemin] [expression]

Par exemple, vous souhaitez rechercher un fichier appelé **notes.txt** dans le répertoire **personnel** et ses sous-dossiers :

find /home -name notes.txt

Voici d'autres variantes de l'utilisation de **find** :

- **find -name filename.txt** pour rechercher des fichiers dans le répertoire actuel.
- **find ./ -type d -nom répertoire** pour rechercher des répertoires.

14. Commande grep

Une autre commande de base sur la liste est **grep** ou **impression globale d'expressions régulières**. Elle vous permet de trouver un mot en parcourant tous les textes d'un fichier spécifique.

Lorsque elle trouve une correspondance, cette commande affiche toutes les lignes qui contiennent le mot spécifique. Cette commande permet aussi de filtrer les fichiers journaux volumineux.

Par exemple, vous souhaitez rechercher le mot **bleu** dans le fichier **notepad.txt** :

grep blue notepad.txt

La sortie de la commande affichera les lignes qui contiennent **bleu**.

15. Commande df

La **commande df** permet de connaître l'utilisation de l'espace disque du système, en pourcentage et en kilo-octets (Ko). Voici la syntaxe générale :

df [options] [fichier]

Par exemple, entrez la commande suivante si vous voulez voir l'utilisation de l'espace disque du répertoire actuel dans un format lisible par l'homme :

df -h

Voici quelques options acceptables à utiliser :

- **df -m** affiche des informations sur l'utilisation du système de fichiers en **Mo**.
- **df -k** affiche l'utilisation du système de fichiers en **Ko**.
- **df -T** affiche le **type de** système de fichiers dans une nouvelle colonne.

16. Commande du

Si vous souhaitez vérifier l'espace occupé par un fichier ou un répertoire, utilisez la commande **du**. Vous pouvez exécuter cette commande pour identifier la partie du système qui utilise l'espace de stockage de manière excessive.

N'oubliez pas que vous devez spécifier le chemin d'accès au répertoire lorsque vous utilisez la commande **du**. **Par exemple**, pour vérifier **/home/user/Documents**, entrez :
du /home/user/Documents

L'ajout d'un drapeau à la commande **du** modifie l'opération, par exemple :

- **-s** offre la taille totale d'un dossier spécifié.
- **-m** fournit des informations sur les dossiers et les fichiers en **Mo**
- **k** affiche les informations en **KB**.
- **-h** informe de la date de dernière modification des dossiers et fichiers affichés.

17. Commande head

La commande **head** permet de visualiser les dix premières lignes d'un texte. L'ajout d'une option permet de modifier le nombre de lignes affichées. La commande **head** est également utilisée pour envoyer des données à la ligne de commande.

Voici la syntaxe générale :

head [option] [fichier]

Par exemple, vous souhaitez afficher les dix premières lignes du fichier **note.txt**, situé dans le répertoire actuel :
note de tête.txt

Vous trouverez ci-dessous quelques options que vous pouvez ajouter :

- **-n** ou **-lines** affiche le premier nombre personnalisé de lignes. Par exemple, entrez **head -n 5 nomfichier.txt** pour afficher les cinq premières lignes de **nomfichier.txt**.
- **-c** ou **-bytes** affiche le premier nombre personnalisé d'octets de chaque fichier.
- **-q** ou **-quiet** n'imprime pas les en-têtes spécifiant le nom du fichier.

18. Commande tail

La **commande tail** affiche les dix dernières lignes d'un fichier. Elle permet aux utilisateurs de vérifier si un fichier contient de nouvelles données ou de lire les messages d'erreur.

Voici le format général :

tail [option] [fichier]

Par exemple, vous souhaitez afficher les dix dernières lignes du fichier **colors.txt** :

tail -n colors.txt

19. Commande diff

Abréviation de différence, la commande **diff** compare deux contenus d'un fichier ligne par ligne. Après les avoir analysés, elle affiche les parties qui ne correspondent pas.

Les programmeurs utilisent souvent la commande **diff** pour modifier un programme au lieu de réécrire tout le code source.

Voici le format général :

diff [option] file1 file2

Par exemple, vous souhaitez comparer deux fichiers texte – **note.txt** et **note_update.txt** :

diff note.txt note_update.txt

Voici quelques options acceptables à ajouter :

- **-c** affiche la différence entre deux fichiers dans un formulaire contextuel.
- **-u** affiche la sortie sans les informations redondantes.
- **-i** rend la commande **diff** insensible à la casse.

20. Commande tar

La **commande tar** permet d'archiver plusieurs fichiers dans un fichier **TAR** – un format Linux commun similaire à **ZIP**, avec une compression optionnelle.

Voici la syntaxe de base :

tar [options] [fichier_archive] [fichier ou répertoire à archiver]

Par exemple, vous souhaitez créer une nouvelle archive **TAR** nommée **newarchive.tar** dans le répertoire **/home/user/Documents** :

tar -cvf newarchive.tar /home/user/Documents

La commande **tar** accepte de nombreuses options, telles que :

- **-x** extrait un fichier.
- **-t** répertorie le contenu d'un fichier.
- **-u** archive et ajoute à un fichier d'archive existant.

Consultez les **exemples pratiques** pour en savoir plus sur les autres fonctions.

21. Commande **chmod**

chmod est une commande courante qui modifie les droits de lecture, d'écriture et d'exécution d'un fichier ou d'un répertoire. Sous Linux, chaque fichier est associé à trois classes d'utilisateurs : **propriétaire**, **membre d'un groupe** et **autres**.

Voici la syntaxe de base :

chmod [option] [permission] [nom_du_fichier]

Par exemple, le propriétaire est actuellement le seul utilisateur à disposer de toutes les autorisations pour modifier **note.txt**. Pour permettre aux membres du groupe et à d'autres personnes de lire, d'écrire et d'exécuter le fichier, attribuez-lui le type d'autorisation **-rwxrwxrwx**, dont la valeur numérique est **777** :

chmod 777 note.txt

Cette commande comporte de nombreuses options, notamment

- **-c** ou **-changes** affiche des informations lorsqu'une modification est apportée.
- **-f** ou **-silent** supprime les messages d'erreur.
- **-v** ou **-verbose** affiche un diagnostic pour chaque fichier traité.

22. Commande chown

La [commande chown](#) vous permet de modifier la propriété d'un fichier, d'un répertoire ou d'un lien symbolique en faveur d'un nom d'utilisateur spécifique. Voici le format de base :

chown [option] propriétaire[:groupe] fichier(s)

Par exemple, vous voulez faire de **linuxuser2** le propriétaire de **filename.txt** :

chown linuxuser2 filename.txt

23. Commande jobs

Un job est un processus que l'interpréteur de commandes lance. La commande **jobs** affiche tous les processus en cours d'exécution ainsi que leur état. N'oubliez pas que cette commande n'est disponible que dans les shells **csh**, **bash**, **tcsh** et **ksh**.

Il s'agit de la syntaxe de base :

jobs [options] jobID

Pour vérifier l'état des travaux dans le shell actuel, il suffit d'entrer **jobs** dans l'interface CLI.

Voici quelques options que vous pouvez utiliser :

- **-l** dresse la liste des identifiants de processus et des informations les concernant.
- **-n** dresse la liste des travaux dont l'état a changé depuis la dernière notification.
- **-p** répertorie uniquement les identifiants de processus.

24. Commande kill

Utilisez la **commande kill** pour mettre fin manuellement à un programme qui ne répond pas. Elle signale les applications qui se comportent mal et leur demande de fermer leurs processus.

Pour tuer un programme, vous devez connaître son numéro d'identification de processus (PID). Si vous ne connaissez pas le PID, exécutez la commande suivante :

ps ux

Après avoir déterminé le signal à utiliser et le PID du programme, entrez la syntaxe suivante :

kill [signal_option] pid

Il existe **64 signaux** que vous pouvez utiliser, mais ces deux-là sont parmi les plus courants :

- **SIGTERM** demande à un programme d'arrêter de fonctionner et lui donne le temps de sauvegarder tous ses progrès. Le système l'utilisera par défaut si vous ne spécifiez pas le signal lors de l'entrée de la commande kill.
- **SIGKILL** force les programmes à s'arrêter, et vous perdrez les progrès non sauvegardés.

Par exemple, le PID du programme est **63773** et vous voulez le forcer à s'arrêter :

tuer SIGKILL 63773

25. Commande ping

La **commande ping** est l'une des commandes Linux de base les plus utilisées pour vérifier si un réseau ou un serveur est joignable. Elle est également utilisée pour résoudre divers problèmes de connectivité.

Voici le format général :

ping [option] [nom_d'hôte ou_adresse_IP]

Par exemple, vous voulez savoir si vous pouvez vous connecter à **Google** et mesurer son temps de réponse :
ping google.com

26. Commande wget

La ligne de commande Linux vous permet de télécharger des fichiers depuis l'internet à l'aide de la **commande wget**. Elle fonctionne en arrière-plan sans gêner les autres processus en cours.

La commande **wget** permet de récupérer des fichiers en utilisant les protocoles HTTP, HTTPS et FTP. Elle peut effectuer des téléchargements récursifs, qui transfèrent des parties de sites web en suivant les structures de répertoires et les liens, créant ainsi des versions locales des pages web.

Pour l'utiliser, entrez la commande suivante :

wget [option] [url]

Par exemple, entrez la commande suivante pour télécharger la dernière version de WordPress :

wget https://wordpress.org/latest.zip

27. Commande uname

La commande **uname** ou **unix name** permet d'obtenir des informations détaillées sur votre système Linux et votre matériel. Ces informations comprennent le nom de la

machine, le système d'exploitation et le noyau. Pour exécuter cette commande, il suffit d'entrer **uname** dans votre CLI.

Voici la syntaxe de base :

uname [option]

Ce sont les options acceptables à utiliser :

- **-a** affiche toutes les informations sur le système.
- **-s** imprime le nom du noyau.
- **-n** imprime le nom d'hôte du nœud du système.

28. Commande top

La **commande top** dans le terminal Linux affiche tous les processus en cours et une vue dynamique en temps réel du système actuel. Elle résume l'utilisation des ressources, de l'unité centrale à l'utilisation de la mémoire.

La commande **top** peut également vous aider à identifier et à mettre fin à un processus qui utilise trop de ressources système.

Pour exécuter la commande, il suffit d'entrer **top** dans le CLI.

29. Commande history

Avec la commande **history**, le système listera jusqu'à 500 commandes précédemment exécutées, ce qui vous permettra de les réutiliser sans avoir à les saisir à nouveau. N'oubliez pas que seuls les utilisateurs disposant des privilèges **sudo** peuvent exécuter cette commande. Le fonctionnement de cet utilitaire dépend également de l'interpréteur de commandes Linux que vous utilisez.

Pour l'exécuter, entrez la commande ci-dessous :

history [option]

Cette commande prend en charge de nombreuses options, telles que :

- **-c** efface la liste complète de l'historique.
- **-d offset** supprime l'entrée de l'historique à la position **OFFSET**.
- **-a** ajoute des lignes d'historique.

30. Commande man

La commande **man** fournit un manuel d'utilisation de toutes les commandes ou utilitaires que vous pouvez exécuter dans le terminal, y compris le nom, la description et les options.

Il se compose de neuf sections :

- Programmes exécutables ou commandes shell
- Appels système
- Appels à la bibliothèque
- Jeux
- Dossiers spéciaux
- Formats de fichiers et conventions
- Commandes d'administration du système
- Routines du noyau
- Divers

Pour afficher le manuel complet, entrez :

man [nom_de_la_commande]

Par exemple, vous souhaitez accéder au manuel de la commande **ls** :

man ls

Entrez cette commande si vous souhaitez spécifier la section affichée :

**man [option] [numéro_de_section]
[nom_de_la_commande]**

Par exemple, vous souhaitez consulter la section 2 du manuel de la commande **ls** :

homme 2 ls

31. Commande echo

La commande **echo** est un utilitaire intégré qui affiche une ligne de texte ou une chaîne de caractères sur la sortie standard. Voici la syntaxe de base :

echo [option] [chaîne]

Par exemple, vous pouvez afficher le texte **Tutoriels Hostinger** en entrant :

echo « Tutoriels Hostinger »

Cette commande prend en charge de nombreuses options, telles que :

- **-n** affiche la sortie sans la nouvelle ligne de fin.
- **-e** permet d'interpréter les barres obliques inverses suivantes :
- **\a** joue une alerte sonore.
- **\b** supprime les espaces entre les textes.
- **\c** ne produit rien d'autre.
- **-E** affiche l'option par défaut et désactive l'interprétation des barres obliques inverses.

32. Commandes zip, unzip

Utilisez la commande **zip** pour compresser vos fichiers dans un fichier **ZIP**, un format universel couramment

utilisé sous Linux. Il peut choisir automatiquement le meilleur taux de compression.

La commande **zip** est également utile pour archiver des fichiers et des répertoires et réduire l'utilisation du disque. Pour l'utiliser, entrez la syntaxe suivante :

zip [options] zipfile file1 file2....

Par exemple, vous avez un fichier nommé **note.txt** que vous voulez compresser dans **archive.zip** dans le répertoire actuel :

zip archive.zip note.txt

En revanche, la [commande unzip](#) extrait les fichiers zippés d'une archive. Voici le format général :

unzip [option] nom_de_fichier.zip

Ainsi, pour décompresser un fichier appelé **archive.zip** dans le répertoire actuel, entrez :
unzip archive.zip

33. Commande hostname

Exécutez la commande **hostname** pour connaître le nom d'hôte du système. Vous pouvez l'exécuter avec ou sans option. Voici la syntaxe générale :

hostname [option]

Il existe de nombreux drapeaux (flags) optionnels à utiliser, notamment :

- **-a** ou **-alias** affiche l'alias du nom d'hôte.
- **-A** ou **-all-fqdns** affiche le nom de domaine entièrement qualifié (FQDN) de la machine.
- **-i** ou **-ip-address** affiche l'adresse IP de la machine.

Par exemple, entrez la commande suivante pour connaître l'adresse IP de votre ordinateur :

hostname -i

34. Commandes **useradd**, **userdel**

Linux est un système multi-utilisateurs, ce qui signifie que plusieurs personnes peuvent l'utiliser simultanément. La commande **useradd** permet de créer un nouveau compte, tandis que la commande **passwd** permet d'ajouter un mot de passe. Seules les personnes disposant des privilèges root ou **sudo** peuvent exécuter la commande **useradd**.

Lorsque vous utilisez la commande **useradd**, elle effectue quelques changements majeurs :

- Modifie les fichiers **/etc/passwd**, **/etc/shadow**, **/etc/group** et **/etc/gshadow** pour les comptes nouvellement créés.
- Crée et alimente un répertoire personnel pour l'utilisateur.
- Définit les autorisations et les propriétaires de fichiers dans le répertoire personnel.

Voici la syntaxe de base :

useradd [option] nom d'utilisateur

Pour définir le mot de passe :

passwd combinaison_mot_de_passe

Par exemple, pour ajouter une nouvelle personne nommée John, entrez simultanément la commande suivante :

useradd John

passwd 123456789

Pour supprimer un compte d'utilisateur, utilisez la commande **userdel** :

userdel nom d'utilisateur

35. Commande **apt-get**

apt-get est un outil de ligne de commande permettant de gérer les bibliothèques APT (Advanced Package Tool) sous Linux. Il vous permet de récupérer des informations et des paquets à partir de sources authentifiées pour gérer, mettre à jour, supprimer et installer des logiciels et leurs dépendances.

L'exécution de la commande **apt-get** nécessite l'utilisation des privilèges **sudo** ou root.

Voici la syntaxe principale :

apt-get [options] (commande)

Voici les commandes les plus courantes que vous pouvez ajouter à **apt-get** :

- **update** synchronise les fichiers du paquet à partir de leurs sources.
- **upgrade** installe la dernière version de tous les paquets installés.
- **check** met à jour le cache des paquets et vérifie les dépendances défectueuses.

36. Commandes nano, vi, jed

Linux permet aux utilisateurs d'éditer et de gérer des fichiers à l'aide d'un éditeur de texte, tel

que **nano**, **vi** ou **jed**. **nano** et **vi** sont fournis avec le système d'exploitation, tandis que **jed** doit être installé.

La **commande nano** désigne des mots-clés et fonctionne avec la plupart des langages. Pour l'utiliser, entrez la commande suivante :

nano [nom de fichier]

vi utilise deux modes de fonctionnement

: **insert** et **command**. **insert** est utilisé pour éditer et créer un fichier texte. En revanche, la **commande** permet

d'effectuer des opérations telles que l'enregistrement, l'ouverture, la copie et le collage d'un fichier.

Pour utiliser **vi** sur un fichier, entrez :

vi [nom de fichier]

jed dispose d'une interface à menu déroulant qui permet aux utilisateurs d'effectuer des actions sans avoir à saisir de combinaisons ou de commandes au clavier.

Comme **vi**, il dispose de modes permettant de charger des modules ou des plugins pour écrire des textes spécifiques.

Pour ouvrir le programme, il suffit d'entrer **jed** dans la ligne de commande.

37. Commandes alias et unalias

La commande **alias** permet de créer un raccourci ayant les mêmes fonctionnalités qu'une commande, un nom de fichier ou un texte. Lorsqu'il est exécuté, il demande à l'interpréteur de commandes de remplacer une chaîne par une autre.

Pour utiliser la commande **alias**, entrez la syntaxe suivante :

alias Nom=Chaîne

Par exemple, vous voulez faire de **k** l'alias de la commande **kill** :

alias k='kill'

En revanche, la commande **unalias** supprime un alias existant.

Voici à quoi ressemble la syntaxe générale :

unalias [nom_alias]

38. Commande su

La commande `switch user` ou **su** vous permet d'exécuter un programme en tant qu'utilisateur différent. Elle modifie le compte administratif dans la session de connexion en cours. Cette commande est particulièrement utile pour accéder au système via **SSH** ou pour utiliser le gestionnaire d'affichage GUI lorsque l'utilisateur root n'est pas disponible.

Voici la syntaxe générale de la commande :

su [options] [nom d'utilisateur [argument]]

Lorsqu'elle est exécutée sans option ni argument, la commande **su** s'exécute avec les privilèges de l'utilisateur root. Elle vous demandera de vous authentifier et d'utiliser temporairement les privilèges **sudo**.

Voici quelques options acceptables à utiliser :

- **-p** ou **-preserve-environment** conserve le même environnement shell, composé de HOME, SHELL, USER et LOGNAME.
- **-s** ou **-shell** vous permet de spécifier un environnement shell différent à exécuter.
- **-l** ou **-login** exécute un script de connexion pour passer à un autre nom d'utilisateur. Pour l'exécuter, vous devez saisir le mot de passe de l'utilisateur.

39. Commande **htop**

La commande **htop** est un programme interactif qui surveille les ressources du système et les processus du serveur en temps réel. Elle est disponible sur la plupart des distributions Linux, et vous pouvez l'installer en utilisant le gestionnaire de paquets par défaut.

Par rapport à la commande **top**, **htop** présente de nombreuses améliorations et fonctionnalités

supplémentaires, telles que l'utilisation de la souris et des indicateurs visuels.

Pour l'utiliser, exécutez la commande suivante :

htop [options]

Vous pouvez également ajouter des options, telles que :

- **-d** ou **-delay** indique le délai entre les mises à jour en dixièmes de secondes.
- **-C** ou **-no-color** active le mode monochrome.
- **-h** ou **-help** affiche le message d'aide et quitte.

40. Commande ps

La commande d'état des processus ou **ps** permet d'obtenir un aperçu de tous les processus en cours d'exécution dans votre système. Les résultats statiques proviennent des fichiers virtuels du système de fichiers **/proc**.

L'exécution de la commande **ps** sans option ni argument permet d'obtenir la liste des processus en cours d'exécution dans l'interpréteur de commandes, ainsi que la liste des processus en cours d'exécution dans l'interpréteur de commandes :

- L'identifiant unique du processus (**PID**)
- Le type de terminal (**TTY**)
- La durée de fonctionnement (**TIME**)
- La commande qui lance le processus (**CMD**)

Voici quelques options acceptables que vous pouvez utiliser :

- **-T** affiche tous les processus associés à la session shell en cours.

- **-u nom d'utilisateur** liste les processus associés à un utilisateur spécifique.
- **-A** ou **-e** affiche tous les processus en cours.

Conseils et astuces

Voici quelques conseils et astuces que vous pouvez utiliser pour gérer le système Linux :

- Entrez la commande **clear** pour nettoyer l'écran du terminal.
- Appuyez sur la touche **Tab** pour remplir automatiquement le formulaire après avoir saisi une commande avec un argument.
- Utilisez **Ctrl + C** pour mettre fin à une commande en cours.
- Appuyez sur **Ctrl + Z** pour mettre en pause une commande en cours.
- Utilisez **Ctrl + S** pour geler temporairement votre terminal.
- Appuyez sur **Ctrl + Q** pour annuler le gel du terminal.
- Utilisez **Ctrl + A** pour vous déplacer au début de la ligne.
- Appuyez sur **Ctrl + E** pour atteindre la fin de la ligne.
- Lorsque vous exécutez plusieurs commandes sur une même ligne, utilisez (;) pour les séparer. Vous pouvez également utiliser **&&** pour n'autoriser l'exécution de la commande suivante que si la précédente a été exécutée avec succès.

Conseil d'expert

Saviez-vous que vous pouviez éditer un fichier texte avec des commandes Linux en utilisant SSH ? Au lieu d'éditer

un fichier localement et de le télécharger via FTP, vous pouvez éditer le fichier instantanément sur votre compte à l'aide de la commande vim ou nano.

Paulius M.

Responsable technique du monitoring de l'hébergement web

Aide-mémoire commandes linux

Utilisez ce tableau pour trouver rapidement la commande que vous recherchez et son fonctionnement

Commande	Explication
sudo	Exécute une commande avec des privilèges administratifs (superutilisateur)
pwd	Affiche le répertoire de travail actuel
cd	Change le répertoire courant

Commande	Explication
ls	Liste les fichiers et répertoires dans le répertoire courant
cat	Concatène et affiche le contenu des fichiers
cp	Copie des fichiers et des répertoires
mv	Déplace ou renomme des fichiers et des répertoires
mkdir	Crée un nouveau répertoire
rmdir	Supprime un répertoire vide
rm	Supprime des fichiers et des répertoires
touch	Crée un fichier vide ou met à jour l'horodatage d'un fichier existant
locate	Recherche des fichiers et des répertoires dans une base de données pré-construite
find	Recherche des fichiers et des répertoires dans un emplacement spécifié
grep	Recherche un motif dans des fichiers ou des flux d'entrée

Commande	Explication
df	Affiche l'utilisation de l'espace disque des systèmes de fichiers
du	Estime l'utilisation de l'espace disque des fichiers et des répertoires
head	Affiche la première partie des fichiers
tail	Affiche la dernière partie des fichiers
diff	Compare les fichiers ligne par ligne
tar	Archive des fichiers et des répertoires dans un seul fichier
chmod	Modifie les permissions des fichiers et des répertoires
chown	Modifie le propriétaire des fichiers et des répertoires
jobs	Liste les tâches actives
kill	Envoie un signal pour terminer des processus
ping	Envoie des requêtes réseau à une adresse IP ou un nom de domaine spécifique

Commande	Explication
wget	Télécharge des fichiers depuis le web
uname	Affiche des informations système
top	Affiche l'utilisation des ressources système en temps réel
history	Affiche l'historique des commandes
man	Affiche les pages de manuel des commandes
echo	Affiche les arguments ou les variables donnés
zip	Crée une archive compressée zip à partir de fichiers
unzip	Extrait des fichiers d'une archive zip
hostname	Affiche ou définit le nom d'hôte du système
useradd	Crée un nouveau compte utilisateur
userdel	Supprime un compte utilisateur
apt-get	Gère les tâches de gestion de paquets
nano	Un éditeur de texte pour

Commande	Explication
	la ligne de commande
vi	Un éditeur de texte pour la ligne de commande
jed	Un éditeur de texte léger pour la ligne de commande
alias	Crée un raccourci ou un alias pour une commande
unalias	Supprime un alias pour une commande

Vous avez besoin d'un aide-mémoire plus avancé ?
Téléchargez-le ci-dessous

Conclusion

Les commandes Linux vous permettent d'effectuer rapidement et efficacement des tâches de base et

d'administration à partir du terminal. Vous disposerez ainsi d'une plus grande souplesse sur votre système et pourrez accéder à des fonctions qui ne sont pas toujours disponibles via une interface utilisateur graphique.

Il est essentiel d'apprendre les commandes de base de Linux pour interagir avec votre machine, surtout si vous utilisez un serveur privé virtuel (**SPV**). Dans la plupart des cas, chaque utilitaire se compose de trois parties :

un **nom de commande**, un **drapeau** ou une **option**, et un **paramètre** ou un **argument**.

Cet article a abordé 40 commandes courantes, telles que **apt-get** pour installer un paquetage, **nano** pour manipuler un fichier, **htop** pour surveiller les processus en cours et **ls** pour afficher un répertoire.

Nous espérons que cet article vous a aidé à apprendre les commandes de base de Linux. Si vous avez des questions ou des suggestions, n'hésitez pas à les laisser dans la section des commentaires ci-dessous.

FAQ sur les commandes Linux

Quelle est la commande de base de Linux ?

L'une des commandes Linux les plus utilisées est la commande cat (abréviation de concatenate). Cette commande vous permet de créer, d'afficher et de combiner des fichiers, ainsi que de rediriger la sortie (le contenu du fichier entier).

Combien de commandes Linux possède-t-il ?

Il existe plus de 100 commandes communes à Linux et à d'autres systèmes d'exploitation similaires.