**35 commandes Linux que vous devez connaitre**

**1. Commande pwd**

Utilisez la commande **pwd** pour trouver le chemin du répertoire de travail (dossier) dans lequel vous êtes actuellement. La commande retournera un chemin absolu (complet), qui est en fait un chemin de tous les répertoires qui commence par une barre oblique **(/)**. Un exemple de chemin absolu est **/home/utilisateur**.

**2. Commande cd**

Pour naviguer dans les fichiers et répertoires de [Linux](https://www.hostinger.fr/tutoriels/commande-tail-de-linux/), utilisez la commande **cd**. Elle nécessite soit le chemin d’accès complet, soit le nom du répertoire, selon le répertoire de travail dans lequel vous vous trouvez.

Disons que vous êtes dans **/home/utilisateur/Documents** et que vous voulez aller dans **Photos**, un sous-répertoire de **Documents**. Pour ce faire, il vous suffit de taper la commande suivante : **cd Photos**.

Un autre scénario est possible si vous voulez passer à un répertoire complètement nouveau, par exemple, **/home/utilisateur/Films**. Dans ce cas, vous devez taper **cd** suivi du chemin absolu du répertoire : **cd /home/utilisateur/Films**.

Il existe quelques raccourcis pour vous aider à naviguer rapidement :

* **cd ..** (avec deux points) pour se déplacer d’un répertoire vers le haut
* **cd** pour aller directement au dossier principal (home)
* **cd-** (avec un tiret) pour passer à votre répertoire précédent

Par ailleurs, le shell de [Linux](https://www.hostinger.fr/tutoriels/commande-shutdown-de-linux/) est sensible à la casse. Vous devez donc taper les noms des répertoires exactement comme ils sont.

**3. Commande ls**

La commande **ls** est utilisée pour visualiser le contenu d’un répertoire. Par défaut, cette commande affichera le contenu de votre répertoire de travail actuel.

Si vous voulez voir le contenu d’autres répertoires, tapez **ls** et ensuite le chemin d’accès du répertoire. Par exemple, tapez **ls /home/utilisateur/Documents** pour voir le contenu de **Documents**.

Il existe des variantes que vous pouvez utiliser avec la commande **ls** :

* **ls -R** énumérera également tous les fichiers dans les sous-répertoires
* **ls -a** affichera les fichiers cachés
* **ls -al** listera les fichiers et les répertoires avec des informations détaillées comme les autorisations, la taille, le propriétaire, etc.

**4. Commande cat**

**cat** (abréviation de concatenate) est l’une des commandes Linux les plus fréquemment utilisées. Elle est utilisée pour lister le contenu d’un fichier sur le résultat standard (sdout). Pour exécuter cette commande, tapez **cat** suivi du nom du fichier et de son extension. Par exemple : **cat fichier.txt**.

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

* **cat > nomDeFichier** crée un nouveau fichier
* **cat nomDeFichier1 nomDeFichier2>nomDeFichier3** joint deux fichiers (1 et 2) et enregistre le résultat de ces derniers dans un nouveau fichier (3)
* pour convertir un fichier en majuscules ou en minuscules, **cat nomDeFichier | tr a-z A-Z >resultat.txt**

**5. Commande cp**

Utilisez la commande **cp** pour copier les fichiers du répertoire actuel dans un autre répertoire. Par exemple, la commande **cp scenery.jpg /home/utilisateur/Photos** créera une copie de **scenery.jpg** (de votre répertoire actuel) dans le répertoire **Photos**.

**6. Commande mv**

L’utilisation principale de la commande **mv** est de déplacer des fichiers, bien qu’elle puisse également être utilisée pour renommer des fichiers.

Les arguments de **mv** sont similaires à ceux de la commande **cp**. Vous devez taper **mv**, le nom du fichier et le répertoire de destination. Par exemple : **mv fichier.txt /home/utilisateur/Documents**.

Pour renommer les fichiers, la commande [Linux](https://www.hostinger.fr/tutoriels/commande-ping-linux-avec-exemples/) est **mv ancien\_nom.ext nouveau\_nom.ext**

**7. Commande mkdir**

Utilisez la commande **mkdir** pour créer un nouveau répertoire – si vous tapez **mkdir Music**, cela créera un répertoire appelé **Music**.

Il existe également des commandes **mkdir** supplémentaires :

* Pour générer un nouveau répertoire à l’intérieur d’un autre répertoire, utilisez cette commande de base de Linux **mkdir Music/Nouveau**
* utiliser l’option **p** (parents) pour créer un répertoire entre deux répertoires existants. Par exemple, **mkdir -p** **Musique/2020/Nouveau** créera le nouveau répertoire « **2020**« .

**8. Commande rmdir**

Si vous avez besoin de supprimer un répertoire, utilisez la commande **rmdir**. Cependant, rmdir ne vous permet de supprimer que les répertoires vides.

**9. Commande rm**

La commande **rm** est utilisée pour supprimer les répertoires et leur contenu. Si vous voulez seulement supprimer le répertoire – comme alternative à **rmdir** – utilisez **rm -r**.

**Note :** Soyez très prudent avec cette commande et vérifiez à nouveau dans quel répertoire vous vous trouvez. Cela effacera tout et il n’y aura pas d’annulation.

**10. Commande touch**

La commande **touch** vous permet de créer un nouveau fichier vierge via la **ligne de commande Linux**. Par exemple, entrez touch **/home/username/Documents/Web.html** pour créer un fichier HTML intitulé **Web** dans le répertoire **Documents**.

**11. Commande locate**

Vous pouvez utiliser cette commande pour **localiser** un fichier, tout comme la commande de recherche dans Windows. De plus, l’utilisation de l’argument **-i** avec cette commande la rendra insensible à la casse, ce qui vous permettra de rechercher un fichier même si vous ne vous souvenez pas de son nom exact.

Pour rechercher un fichier qui contient deux mots ou plus, utilisez un astérisque **(\*)**. Par exemple, la commande « **locate -i school\*note** » permettra de rechercher tout fichier contenant les mots « **school** » et « **note**« , qu’ils soient en majuscules ou en minuscules.

**12. Commande find**

Comme la commande **locate**, l’utilisation de **find** permet également de rechercher des fichiers et des répertoires. La différence est que vous utilisez la commande **find** pour localiser des fichiers dans un répertoire donné.

Par exemple, la commande **find** **/home/ -name notes.txt** permet de rechercher un fichier appelé **notes.txt** dans le répertoire home et ses sous-répertoires.

Il existe d’autres variations dans l’utilisation de **find** :

* Pour trouver des fichiers dans le répertoire actuel, utilisez, **find . -name notes.txt**
* Pour rechercher des répertoires, utilisez, **/ -type d -name notes. txt**

**13. grep command**

Une autre commande de base de [Linux](https://www.hostinger.fr/tutoriels/comment-desactiver-selinux-sur-centos/) qui est sans aucun doute utile pour une utilisation quotidienne est **grep**. Elle vous permet de rechercher tout le texte d’un fichier donné.

Par exemple, **grep blue notepad.txt** recherchera le mot **blue** dans le fichier **notepad**. Les lignes qui contiennent le mot recherché s’afficheront entièrement.

**14. Commande sudo**

Abréviation de « **SuperUser Do**« , cette commande vous permet d’effectuer des tâches qui nécessitent des autorisations administratives ou de root. Cependant, il n’est pas conseillé d’utiliser cette commande pour un usage quotidien car une erreur pourrait facilement se produire si vous avez fait quelque chose de incorrect.

**15. Commande df**

Utilisez la commande **df** pour obtenir un rapport sur l’utilisation de l’espace disque du système, indiquée en pourcentage et en Ko. Si vous voulez voir le rapport en mégaoctets, tapez **df -m**.

**16. Commande du**

Si vous voulez vérifier l’espace occupé par un fichier ou un répertoire, la commande **du** (Disk Usage) est la réponse. Cependant, le résumé de l’utilisation du disque indiquera les numéros de bloc du disque au lieu du format habituel de la taille. Si vous voulez le voir en octets, kilo-octets et méga-octets, ajoutez l’argument **-h** à la ligne de commande.

**17. Commande head**

La commande **head** est utilisée pour visualiser les premières lignes de n’importe quel fichier texte. Par défaut, elle affichera les dix premières lignes, mais vous pouvez modifier ce nombre à votre convenance. Par exemple, si vous ne voulez afficher que les cinq premières lignes, tapez **head -n 5 nomdefichier.ext**.

**18. Commande tail**

Celle-ci a une fonction similaire à celle de la commande head, mais au lieu d’afficher les premières lignes, la commande **tail** affichera les dix dernières lignes d’un fichier texte. Par exemple, **tail -n nomdefichier.ext**.

**19. Commande diff**

Abréviation de différence, la commande **diff** compare le contenu de deux fichiers ligne par ligne. Après avoir analysé les fichiers, elle affiche les lignes qui ne correspondent pas. Les programmeurs utilisent souvent cette commande lorsqu’ils ont besoin d’apporter des modifications au programme au lieu de réécrire l’intégralité du code source.

La forme la plus simple de cette commande est **diff fichier1.ext fichier2.ext**

**20. Commande tar**

La commande **tar** est la commande la plus utilisée pour archiver plusieurs fichiers dans un **tarball** – un format de fichier Linux commun qui est similaire au format zip, avec la compression étant optionnelle.

Cette commande est assez complexe et comporte une longue liste de fonctions telles que l’ajout de nouveaux fichiers dans une archive existante, la liste du contenu d’une archive, l’extraction du contenu d’une archive, et bien d’autres encore. Consultez quelques [exemples pratiques](https://www.linuxtechi.com/17-tar-command-examples-in-linux/) pour en savoir plus sur les autres fonctions.

**21. Commande chmod**

**Chmod** est une autre commande Linux, utilisée pour modifier les permissions de lecture, d’écriture et d’exécution des fichiers et des répertoires. Comme cette commande est assez compliquée, vous pouvez lire le [tutoriel complet](https://www.computerhope.com/unix/uchmod.htm) afin de l’exécuter correctement.

**22. Commande chown**

Sous Linux, tous les fichiers sont la propriété d’un utilisateur spécifique. La commande **chown** vous permet de changer ou de transférer la propriété d’un fichier à un utilisateur spécifique. Par exemple, **chown linuxuser2 fichier.ext** fera de **linuxuser2** le propriétaire du **fichier.ext**.

**23. Commande jobs**

La commande **jobs** affichera tous les **jobs** actuels avec leur statut. Un job est essentiellement un processus qui est lancé par le shell.

**24. Commande kill**

Si vous avez un programme qui ne répond pas, vous pouvez l’arrêter manuellement en utilisant la commande **kill**. Celle-ci enverra un certain signal à l’application qui se comporte mal et lui demandera de s’arrêter.

Il y a un total de [soixante-quatre signaux](https://linoxide.com/linux-how-to/linux-signals-part-1/) que vous pouvez utiliser, mais les gens n’utilisent généralement que deux signaux :

* **SIGTERM (15)** — demande à un programme de s’arrêter de fonctionner et lui donne un peu de temps pour enregistrer tous ses progrès. Si vous ne spécifiez pas le signal lors de la saisie de la commande d’arrêt, ce signal sera utilisé.
* **SIGKILL (9)** — oblige les programmes à s’arrêter immédiatement. Les progrès non sauvegardés seront perdus.

Outre la connaissance des signaux, vous devez également connaître le numéro d’identification du processus (PID) du programme que vous voulez arrêter. Si vous ne connaissez pas le **PID**, il vous suffit d’exécuter la commande **ps ux**.

Après avoir connu le signal que vous voulez utiliser et le PID du programme, entrez la syntaxe suivante :

**kill [option de signal] PID**.

**25. Commande ping**

Utilisez la commande **ping** pour vérifier votre état de connectivité à un serveur. Par exemple, en entrant simplement **ping google.com**, la commande vérifiera si vous êtes en mesure de vous connecter à Google et mesurera également le temps de réponse.

**26. Commande wget**

**Le terminal Linux** est très puissant. Vous pouvez même l’utiliser pour télécharger des fichiers sur Internet à l’aide de la commande **wget**. Pour ce faire, il suffit de taper **wget** suivi du lien de téléchargement.

**27. Commande uname**

La commande **uname**, abréviation de Unix Name, imprimera des informations détaillées sur votre système Linux comme le nom de la machine, le système d’exploitation, le noyau, etc.

**28. Commande top**

Comme un terminal équivalent au gestionnaire de tâches dans Windows, la commande **top** affichera une liste des processus qui sont en cours d’exécution et la quantité de CPU utilisée par chaque processus. Il est très utile de surveiller l’utilisation des ressources du système, en particulier de savoir quel processus doit être arrêté en cas de surconsommation de ressources.

**29. Commande history**

Lorsque vous utilisez Linux depuis un certain temps, vous remarquerez rapidement que vous pouvez exécuter des centaines de commandes chaque jour. Ainsi, l’exécution de la commande **history** est particulièrement utile si vous voulez revoir les commandes que vous avez entrées auparavant.

**30. Commande man**

Confus quant à la fonction de certaines commandes Linux ? Ne vous inquiétez pas, vous pouvez facilement apprendre à les utiliser directement depuis le shell de Linux en utilisant la commande **man**. Par exemple, en entrant la commande **man tail**, vous verrez les instructions manuelles de la commande tail.

**31. Commande echo**

Cette commande est utilisée pour déplacer certaines données dans un fichier. Par exemple, si vous voulez ajouter le texte « Bonjour, je suis John » dans un fichier appelé nom.txt, vous devez taper **echo Bonjour, je suis John >> nom.txt**

**32. Commande zip, unzip**

Utilisez la commande **zip** pour compresser vos fichiers dans une archive zip, et utilisez la commande **unzip** pour extraire les fichiers zippés d’une archive zip.

**33. Commande hostname**

Si vous voulez connaître le nom de votre hôte/réseau, il vous suffit de taper **hostname**. En ajoutant un **-I** à la fin, vous obtiendrez l’adresse IP de votre réseau.

**34. Commande useradd, userdel**

Puisque Linux est un système multi-utilisateurs, cela signifie que plusieurs personnes peuvent interagir avec le même système en même temps. **useradd** est utilisé pour créer un nouvel utilisateur, tandis que **passwd** est l’ajout d’un mot de passe au compte de cet utilisateur. Pour ajouter une nouvelle personne nommée John, tapez **useradd John** et ensuite pour ajouter son mot de passe, tapez **passwd 123456789**.

La suppression d’un utilisateur est très similaire à l’ajout d’un nouvel utilisateur. Pour supprimer le compte d’un utilisateur, tapez, **userdel NomUtilisateur**

**35. Commande shutdown**

Le terminal Linux vous permet de tout faire y compris éteindre ou redémarrer votre machine. Pour cela vous allez utiliser la commande **shutdown**. Ce qui est intéressant avec cette commande c’est que vous pouvez éteindre/redémarrer votre ordinateur immédiatement en utilisant le paramètre « **now**« .

Et vous pouvez aussi programmer l’arrêt de la machine à une **heure précise**.

**Conseils et astuces**

Utilisez la commande **clear** pour nettoyer le terminal s’il est encombré par trop de commandes Linux déjà effectuées.

Essayez le bouton **TAB** pour remplir automatiquement ce que vous tapez. Par exemple, si vous avez besoin de taper Documents, commencez à taper une commande (disons **cd Docu**, puis appuyez sur la touche **TAB**) et le terminal remplira le reste, ce qui vous donnera **cd Documents**.

**Ctrl+C** et **Ctrl+Z** sont utilisés pour arrêter toute commande qui fonctionne actuellement. Ctrl+C arrêtera et terminera la commande, tandis que Ctrl+Z mettra simplement la commande en pause.

Si vous bloquez accidentellement votre terminal en utilisant **Ctrl+S**, il suffit de le débloquer avec **Ctrl+Q**.

**Ctrl+A** vous amène au début de la ligne tandis que **Ctrl+E** vous amène à la fin.

Vous pouvez exécuter plusieurs commandes en une seule en utilisant « **;** » pour les séparer. Par exemple, **Commande1; Commande2; Commande3**. Ou utilisez **&&** si vous souhaitez que la commande suivante ne soit exécutée que lorsque la première a réussi.

**Pour résumer**

Les commandes Linux de base aident les utilisateurs à exécuter des tâches facilement et efficacement. Il est possible que cela prenne un certain temps pour mémoriser certaines commandes de base, mais rien n’est impossible avec beaucoup de pratique.

En fin de compte, il vous sera sans aucun doute bénéfique de connaître et de maîtriser ces commandes Linux de base. Bonne chance !