

LINUX

Ing Abousidikou ODJO
odjosidik0@gmail.com

INTRODUCTION

- Comprendre l'histoire et la philosophie de Linux
 - Linux a été créé en 1991 par Linus Torvalds, alors étudiant à l'Université d'Helsinki en Finlande. Linus cherchait à créer un système d'exploitation librement accessible, inspiré par Minix, un système d'enseignement petit mais limité conçu par Andrew Tanenbaum. Il voulait améliorer Minix pour répondre à ses propres besoins sans les restrictions de licence.
 - Le 25 août 1991, Linus a annoncé sur le newsgroup `comp.os.minix` qu'il travaillait sur un système d'exploitation libre, une annonce qui a marqué le début officiel du projet Linux. La première version, Linux 0.01, a été rendue publique plus tard cette année-là, et bien qu'elle ait été assez rudimentaire, elle a attiré l'attention de nombreux développeurs à travers le monde.
 - Logiciel Libre: Le logiciel libre se réfère à la liberté des utilisateurs de faire fonctionner, copier, distribuer, étudier, modifier et améliorer le logiciel.
 - Open Source: L'open source met l'accent sur les avantages pratiques de partager le code ouvertement pour développer des logiciels de meilleure qualité.

INTRODUCTION: OS

- Les concepts de base d'un système d'exploitation (OS) sont fondamentaux pour comprendre son fonctionnement:
 - Gestion des ressources : L'OS gère les ressources matérielles telles que le processeur, la mémoire, le stockage et les périphériques d'entrée/sortie. Il attribue ces ressources aux différents processus de manière efficace.
 - Interface utilisateur : L'OS fournit une interface permettant aux utilisateurs d'interagir avec le système. Cela peut être une interface en ligne de commande (CLI) ou une interface graphique utilisateur (GUI).
 - Gestion des fichiers : L'OS gère les fichiers sur le système de fichiers, en permettant leur création, leur modification, leur suppression et leur organisation dans une hiérarchie de répertoires.
 - Gestion des processus : L'OS gère l'exécution des processus. Un processus peut être une application ou un programme en cours d'exécution. L'OS attribue du temps processeur aux processus, gère leur ordonnancement et leur accès aux ressources système.
 - Gestion de la mémoire : L'OS contrôle l'allocation et la libération de la mémoire système. Cela garantit qu'il y a suffisamment de mémoire disponible pour les processus en cours d'exécution et qu'ils ne se corrompent pas mutuellement.

INTRODUCTION: OS

- Les concepts de base d'un système d'exploitation (OS) sont fondamentaux pour comprendre son fonctionnement:
 - Communication inter-processus : L'OS facilite la communication entre les processus en fournissant des mécanismes tels que les tubes, les files d'attente, les signaux, etc.
 - Gestion des périphériques : L'OS contrôle les périphériques matériels tels que les imprimantes, les claviers, les souris, etc. Il fournit des pilotes de périphériques pour permettre aux périphériques de communiquer avec le système.
 - Sécurité et autorisation : L'OS garantit la sécurité du système en contrôlant l'accès aux ressources et en imposant des politiques de sécurité telles que l'authentification des utilisateurs et les autorisations d'accès aux fichiers.
 - Gestion des erreurs : L'OS gère les erreurs matérielles et logicielles en fournissant des mécanismes de détection et de récupération des erreurs, ainsi que des journaux pour le diagnostic.

INTRODUCTION: OS

- Différences entre Linux et d'autres systèmes d'exploitation
 - Licence et modèle de développement :
 - Linux est distribué sous la licence GNU GPL (General Public License), ce qui signifie qu'il est open source et que son code source est accessible à tous pour examen, modification et distribution.
 - Les autres systèmes d'exploitation, tels que Windows et macOS, sont généralement des logiciels propriétaires avec des restrictions sur leur utilisation, leur modification et leur distribution.
 - Noyau monolithique vs micro-noyau :
 - Linux utilise un noyau monolithique, où la majeure partie du système d'exploitation s'exécute en mode noyau, ce qui signifie qu'il est plus intégré et offre généralement de meilleures performances.
 - Certains autres systèmes d'exploitation, comme QNX et MINIX, utilisent des micro-noyaux, où une partie des fonctionnalités du système d'exploitation est exécutée en mode utilisateur, ce qui peut offrir plus de stabilité mais avec une légère perte de performances.

INTRODUCTION: OS

- Différences entre Linux et d'autres systèmes d'exploitation
 - Variété de distributions :
 - Linux est disponible sous forme de nombreuses distributions différentes, chacune avec ses propres choix de logiciels, configurations et philosophies de conception. Ces distributions peuvent être spécialisées pour différents cas d'utilisation, tels que les serveurs, les postes de travail, les appareils mobiles, etc.
 - Les autres systèmes d'exploitation sont généralement disponibles dans des versions standardisées avec moins de variations entre elles.
 - Système de gestion de paquets :
 - La plupart des distributions Linux utilisent un système de gestion de paquets, tel que apt (Debian/Ubuntu), yum/dnf (Red Hat/Fedora), pacman (Arch Linux), etc., qui simplifie l'installation, la mise à jour et la suppression de logiciels.
 - Les autres systèmes d'exploitation peuvent avoir des mécanismes différents pour gérer les logiciels, tels que les installateurs autonomes ou les boutiques d'applications propriétaires.

Distributions populaires

- Les distributions Linux, souvent appelées "distros", sont des variantes du système d'exploitation Linux qui incluent le noyau Linux ainsi qu'un ensemble de logiciels applicatifs.
- Chaque distribution peut varier grandement en termes de gestion des paquets, de configuration par défaut, de philosophie de design et de public cible.
- **Ubuntu**
 - Ubuntu est l'une des distributions Linux les plus populaires, surtout parmi les nouveaux utilisateurs de Linux. Développée par la société Canonical Ltd., Ubuntu est basée sur Debian et est conçue pour offrir une expérience utilisateur simplifiée. Elle est bien connue pour sa régularité de publication, avec de nouvelles versions tous les six mois, et des versions LTS (Long Term Support) tous les deux ans, qui sont supportées pendant cinq ans.
 - Ubuntu est largement utilisé sur les ordinateurs personnels, les serveurs et récemment dans le cloud. Elle est également populaire pour le développement de logiciels.

Distributions populaires

- **Debian**

- Debian est connue pour son adhérence stricte aux principes du logiciel libre, sa stabilité et sa sécurité. C'est une distribution qui privilégie la stabilité et la sécurité, ce qui entraîne souvent des cycles de publication plus longs.
- Debian est utilisée tant sur les serveurs que sur les ordinateurs personnels et est également la base de nombreuses autres distributions, y compris Ubuntu et Mint.

- **Fedora**

- Fedora est une distribution communautaire sponsorisée par Red Hat et sert de banc d'essai pour les technologies qui sont ensuite intégrées dans Red Hat Enterprise Linux. Fedora est connue pour être à la pointe de la technologie, adoptant rapidement les nouvelles technologies.
- Elle est souvent utilisée par les développeurs de logiciels qui souhaitent travailler avec les dernières technologies open source.

Distributions populaires

- **CentOS**

- CentOS (Community ENTerprise Operating System) est une distribution qui est dérivée de Red Hat Enterprise Linux (RHEL). Avec la création de CentOS Stream, elle se positionne désormais comme un système précurseur à RHEL, offrant un aperçu des prochaines fonctionnalités RHEL.
- Principalement utilisé sur les serveurs, CentOS est très populaire dans le monde de l'entreprise en raison de sa stabilité et de sa longue durée de support.

- **Arch Linux**

- Arch Linux se distingue par sa philosophie KISS ("Keep It Simple, Stupid"). Elle est conçue pour être légère et flexible. Arch est une distribution rolling release, ce qui signifie que les mises à jour sont continues et qu'il n'y a pas de versions majeures.
- Arch est particulièrement populaire parmi les utilisateurs Linux avancés qui veulent personnaliser leur système d'exploitation jusqu'au moindre détail.

Distributions populaires

- **Linux Mint**

- Linux Mint est dérivée de Ubuntu et est connue pour être particulièrement conviviale, notamment pour les utilisateurs qui migrent de Windows. Elle offre une expérience utilisateur polie avec un environnement de bureau par défaut Cinnamon qui est à la fois esthétique et fonctionnel.
- Elle est particulièrement adaptée aux débutants et à ceux qui cherchent une distribution "prête à l'emploi" pour un usage quotidien sur un ordinateur personnel.

- **Kali Linux**

- Kali Linux est une distribution basée sur Debian, développée pour l'analyse de sécurité avancée et les tests de pénétration. Elle est équipée d'une grande variété d'outils destinés à l'audit de sécurité et au hacking éthique.
 - Très appréciée par les professionnels de la sécurité informatique et les chercheurs en sécurité.
- Selon DistroWatch, il y a environ 500-600 distributions.

Distributions populaires

- **Linux Mint :**

- Linux Mint est une distribution basée sur Ubuntu et Debian, conçue pour être conviviale et élégante. Elle est livrée avec un ensemble complet de logiciels préinstallés et des environnements de bureau personnalisés, notamment Cinnamon et MATE.

- **Parrot Security OS**

- Parrot Security OS est une distribution Linux basée sur Debian, conçue spécifiquement pour les professionnels de la sécurité informatique, les chercheurs en sécurité et les tests d'intrusion.

Cas d'usage de Linux dans l'industrie et chez les particuliers.

- Dans l'industrie :
 - **Serveurs Web et Infrastructures Cloud** : Linux est le système d'exploitation le plus utilisé sur les serveurs web. Des géants comme Google, Amazon, Facebook et Twitter utilisent massivement des serveurs Linux pour fournir des services en ligne à grande échelle.
 - **Superordinateurs et Calcul Haute Performance** : De nombreux supercalculateurs et clusters de calcul haute performance (HPC) fonctionnent sous Linux en raison de sa stabilité, de sa performance et de sa capacité à évoluer pour répondre aux besoins croissants de calcul intensif.
 - **Systèmes Embarqués** : Linux est largement utilisé dans les systèmes embarqués, tels que les routeurs, les appareils IoT (Internet des Objets), les systèmes de contrôle industriel, les appareils médicaux, les téléviseurs intelligents, et bien d'autres, en raison de sa modularité et de sa personnalisation.
 - **Développement de Logiciels** : Les développeurs de logiciels utilisent fréquemment Linux comme environnement de développement en raison de son support natif pour de nombreux langages de programmation, de ses outils de développement puissants et de sa facilité d'automatisation des tâches.
 - **Administration Système et Réseaux** : Les professionnels de l'informatique utilisent Linux pour gérer les réseaux, les serveurs, les bases de données, les pare-feux, les services web, et d'autres infrastructures informatiques en raison de sa fiabilité, de sa sécurité et de sa flexibilité.

Cas d'usage de Linux dans l'industrie et chez les particuliers.

- Chez les Particuliers :
 - **Ordinateurs Personnels** : De plus en plus de particuliers utilisent Linux comme alternative aux systèmes d'exploitation propriétaires comme Windows et macOS en raison de sa stabilité, de sa sécurité, de sa personnalisation et de son coût nul. Des distributions conviviales comme Ubuntu, Linux Mint et Elementary OS rendent l'expérience utilisateur de Linux accessible à tous.
 - **Divertissement** : Linux est également utilisé pour le divertissement à domicile, comme la lecture de médias, le jeu vidéo (via des plates-formes comme Steam et des distributions comme SteamOS), et la création de serveurs de médias domestiques.
 - **Éducation** : Linux est largement utilisé dans les environnements éducatifs pour enseigner les principes fondamentaux de l'informatique, ainsi que pour fournir un accès à des logiciels éducatifs et à des ressources d'apprentissage en ligne.
 - **IoT et Appareils Embarqués** : Les particuliers utilisent Linux dans des appareils IoT comme les Raspberry Pi, les NAS (Network Attached Storage), les routeurs personnalisés et les consoles de jeux rétro, entre autres.

Installation de Linux

- Savoir préparer une machine pour l'installation de Linux.

What's new	System requirements	How to install
	✓ 2 GHz dual-core processor or better	
	✓ 4 GB system memory	
	✓ 25 GB of free hard drive space	
	✓ Either a USB port or a DVD drive for the installer media	
	✓ Internet access is helpful	

Installation de Linux

- Etapes

- ② Download an Ubuntu Image
- ③ Create a Bootable USB stick
- ④ Boot from USB flash drive
- ⑤ Installation Setup
- ⑥ Type of installation
- ⑦ Create Your Login Details
- ⑧ Choose your Location
- ⑨ Ready to install
- ⑩ Complete the Installation
- ⑪ Don't forget to Update!
- ⑫ You've installed Ubuntu!
- ⑬ (Additional) Installing Ubuntu alongside Windows with BitLocker

Installation de Linux

- Bootage de clé



Installation de Linux

- ...

Installation de Linux: Partitionnement

- Voici une méthode de partitionnement typique pour une installation standard :
 - Partition EFI (pour les systèmes UEFI) : Si votre système utilise l'UEFI, vous aurez besoin d'une partition EFI. Elle doit être formatée en FAT32 et avoir une taille d'au moins 100 Mo.
 - Partition racine (/) : C'est la partition principale où le système d'exploitation sera installé. Elle contient tous les fichiers système. Une taille recommandée est d'au moins 20 Go.
 - Partition swap (optionnelle) : La partition swap est utilisée comme espace d'échange pour la mémoire virtuelle. Sa taille est souvent égale ou double à la taille de la mémoire RAM de votre système, mais cela dépend de vos besoins.
 - Partition /home (optionnelle) : La partition /home contient les répertoires personnels des utilisateurs, y compris les documents, les images, etc. Il est recommandé de la créer séparément pour simplifier les sauvegardes et les réinstallations du système.

Installation de Linux:

1. Installation pas à pas d'Ubuntu :
 - 1.1. Démarrage à partir du média d'installation : Insérez votre clé USB bootable Ubuntu dans votre ordinateur et redémarrez-le. Assurez-vous que votre BIOS ou UEFI est configuré pour démarrer à partir du support USB.
 - 1.2. Choisir la langue : Sélectionnez la langue dans laquelle vous souhaitez installer Ubuntu et cliquez sur "Installer Ubuntu".
 - 1.3. Connexion Internet : Si vous êtes connecté à Internet, vous pouvez choisir de télécharger les mises à jour et les logiciels tiers pendant l'installation. Sinon, vous pouvez les installer ultérieurement.
 - 1.4. Sélectionner l'espace disque : Lorsque vous arrivez à l'étape "Installation type", choisissez "Something else" pour un partitionnement personnalisé.
 - 1.5. Configurer les partitions : Créez les partitions nécessaires en utilisant les tailles recommandées. Assurez-vous de sélectionner la bonne partition de démarrage (EFI ou BIOS) si nécessaire.
 - 1.6. Sélectionner le point de montage : Pour chaque partition, choisissez le point de montage approprié (/ , /home, swap, etc.).
 - 1.7. Confirmer les changements : Une fois les partitions configurées, continuez le processus d'installation. Vous serez invité à confirmer les changements sur le disque.
 - 1.8. Configurer le reste de l'installation : Suivez les instructions à l'écran pour configurer la langue, le fuseau horaire, le clavier, etc. Créez un compte utilisateur et un mot de passe.
 - 1.9. Terminer l'installation : Une fois l'installation terminée, redémarrez votre ordinateur. Retirez le support d'installation lorsque vous y êtes invité.

Installation de Linux: Configuration initiale

Une fois Ubuntu installé, vous devrez effectuer quelques configurations initiales :

1. Mises à jour système : Ouvrez le gestionnaire de mises à jour et installez toutes les mises à jour disponibles pour votre système.
2. Configuration du réseau : Configurez votre connexion Internet si cela n'a pas été fait pendant l'installation.
3. Installation de pilotes (le cas échéant) : Si votre matériel nécessite des pilotes propriétaires, comme les pilotes graphiques, installez-les via le gestionnaire de pilotes additionnels.
4. Configuration des périphériques : Assurez-vous que tous vos périphériques fonctionnent correctement. Si nécessaire, configurez-les via les paramètres système.
5. Personnalisation : Personnalisez votre bureau et vos paramètres système selon vos préférences.

Bases de la ligne de commande

- Le terminal est une interface en ligne de commande qui permet aux utilisateurs d'interagir avec un système d'exploitation en saisissant des commandes texte.
- Il est souvent utilisé pour effectuer des tâches système, lancer des programmes, gérer des fichiers, et bien plus encore.
- Les interpréteurs de commandes, tels que Bash et Zsh, sont des programmes qui exécutent ces commandes et fournissent un environnement pour les utilisateurs pour travailler avec le système.

Bash (Bourne Again Shell) :

Bash est l'interpréteur de commandes par défaut sur la plupart des systèmes Linux et Unix. Il est largement utilisé en raison de sa compatibilité, de sa robustesse et de sa puissance. Voici quelques caractéristiques de Bash :

- **Compatibilité** : Bash est compatible avec la syntaxe Bourne Shell (sh) tout en ajoutant de nombreuses fonctionnalités supplémentaires.
- **Personnalisation** : Les utilisateurs peuvent personnaliser leur environnement Bash en créant des fichiers de configuration comme `.bashrc` et `.bash_profile`.
- **Scripts** : Bash est largement utilisé pour écrire des scripts shell pour l'automatisation de tâches système.

Bases de la ligne de commande

Zsh (Z Shell) :

Zsh est un interpréteur de commandes plus moderne et plus avancé que Bash. Il offre de nombreuses fonctionnalités supplémentaires et une interface utilisateur améliorée. Voici quelques caractéristiques de Zsh :

- Autocomplétion : Zsh offre une fonctionnalité d'autocomplétion avancée qui suggère automatiquement des commandes, des options et des fichiers lors de la saisie.
- Thèmes et plugins : Zsh prend en charge les thèmes et les plugins, ce qui permet aux utilisateurs de personnaliser davantage leur environnement.
- Gestion des tâches : Zsh offre des fonctionnalités avancées de gestion des tâches, telles que la suspension et la reprise de processus en arrière-plan.

Choix entre Bash et Zsh :

Le choix entre Bash et Zsh dépend des préférences personnelles de l'utilisateur et des besoins spécifiques. Bash est plus couramment utilisé et est généralement suffisant pour la plupart des tâches de ligne de commande. Zsh, en revanche, offre une expérience utilisateur plus avancée avec ses fonctionnalités supplémentaires. Certains utilisateurs préfèrent également Zsh pour son autocomplétion plus puissante et ses thèmes personnalisables.

Commande: ls

- La commande `ls` est l'une des commandes les plus couramment utilisées dans un terminal Linux.
- Elle est utilisée pour lister les fichiers et les répertoires dans le répertoire courant ou dans un répertoire spécifié.
- Voici quelques utilisations courantes de la commande `ls` :
 - Lister les fichiers et répertoires dans le répertoire courant:: `ls`
 - Lister les fichiers et répertoires dans un répertoire spécifié :: `ls /chemin`
 - chemin absolue
 - chemin relatif
 - Afficher les fichiers et répertoires avec des détails (permissions, propriétaire, taille, date de modification, etc.) :: `ls -l`
 - Permissions : Les 10 premiers caractères de chaque ligne indiquent les permissions du fichier ou du répertoire. Par exemple, `-rw-r--r--` signifie que le fichier est accessible en lecture et écriture pour le propriétaire, et en lecture seule pour les autres.
 - Nombre de liens : Indique le nombre de liens vers le fichier ou le répertoire.
 - Propriétaire : Le nom de l'utilisateur propriétaire du fichier ou du répertoire.
 - Groupe : Le nom du groupe auquel appartient le fichier ou le répertoire.
 - Taille : La taille du fichier en octets.
 - Date de modification : Indique la date et l'heure de la dernière modification du fichier ou du répertoire.
 - Nom : Le nom du fichier ou du répertoire.

Commande: ls

- Voici quelques utilisations courantes de la commande `ls` :
 - Afficher tous les fichiers, y compris les fichiers cachés (ceux dont le nom commence par un point) :: `ls -a`
 - Afficher les fichiers et répertoires dans un format détaillé avec les fichiers cachés :: ...
 - Trier les fichiers et répertoires par taille :: `ls -S`
 - Trier les fichiers et répertoires par date de modification (du plus récent au plus ancien) :: `ls -t`
 - Afficher les fichiers et répertoires récursivement dans tous les sous-répertoires :: `ls -R`
 - Utiliser une couleur pour différencier les types de fichiers (exécutable, répertoire, lien symbolique, etc.) :: `ls --color=auto`
 - Afficher l'aide pour la commande `ls` :: `man ls`

Commande: cd

- La commande `cd` est utilisée pour changer le répertoire courant dans un terminal Linux.
- Voici quelques utilisations courantes de la commande `cd` :
 - Changer le répertoire courant vers un répertoire spécifié : `cd /chemin/vers/le/repertoire`
 - Revenir au répertoire parent : `cd ..`
 - Aller au répertoire utilisateur : `cd ~` ou `cd`
 - Aller au répertoire précédent : `cd -`
 - Utiliser un raccourci pour un répertoire :
 - Si vous utilisez souvent un répertoire spécifique, vous pouvez créer un alias dans votre fichier de configuration (comme `.bashrc` ou `.zshrc`) : `alias monrepertoire='cd /chemin/vers/le/repertoire'`

Commande: env

- La commande env est utilisée pour afficher les variables d'environnement actuellement définies dans votre session.
- Les variables d'environnement sont des variables globales qui sont disponibles pour tous les processus s'exécutant dans la session courante.
 - Afficher toutes les variables d'environnement : env
 - Afficher la valeur d'une variable d'environnement spécifique : env nom_variable
- Modifier l'environnement d'une commande spécifique : env VARIABLE=valeur commande

Commande: mkdir

- La commande mkdir est utilisée pour créer un nouveau répertoire (aussi appelé dossier) dans le système de fichiers.
- Voici comment l'utiliser :
 - Créer un répertoire dans le répertoire courant : `mkdir nom_du_repertoire`
 - Créer plusieurs répertoires en une seule fois : `mkdir nom_repertoire1 nom_repertoire2 nom_repertoire3`
 - Créer un répertoire avec un chemin spécifique : `mkdir chemin/vers/le/repertoire`
 - Créer plusieurs répertoires avec des sous-répertoires : `mkdir -p chemin/vers/repertoire1/sous_repertoire1 chemin/vers/repertoire2/sous_repertoire2`
 - Créer un répertoire avec des permissions spécifiques : `mkdir -m mode nom_du_repertoire`
 - Créer un répertoire interactif : `mkdir -i nom_du_repertoire`

Commande: touch

- La commande touch est utilisée pour créer un nouveau fichier vide dans le système de fichiers Linux ou pour mettre à jour l'horodatage d'un fichier existant.
- Voici comment utiliser la commande touch :
 - Créer un nouveau fichier vide : touch nom_du_fichier
 - Créer plusieurs fichiers vides en une seule fois : touch fichier1 fichier2 fichier3
 - Créer un fichier avec une extension spécifique : touch fichier.txt

Commande: cp

- La commande cp est une commande utilisée dans les systèmes Unix et Linux pour copier des fichiers et des répertoires d'un emplacement à un autre.
- Voici la syntaxe de base de la commande cp : **cp [OPTIONS] SOURCE DESTINATION**
- **SOURCE** : spécifie le chemin du fichier ou du répertoire source que vous souhaitez copier.
- **DESTINATION** : spécifie le chemin où vous souhaitez copier le fichier ou le répertoire source.
- Voici quelques options couramment utilisées avec la commande cp :
 - -r, --recursive : copie les répertoires de manière récursive, c'est-à-dire qu'elle copie également tous les fichiers et sous-répertoires qu'ils contiennent.
 - -i, --interactive : demande confirmation avant de remplacer les fichiers existants.
 - -v, --verbose : affiche des informations détaillées sur les fichiers copiés.
 - -u, --update : ne copie que les fichiers source qui sont plus récents que les fichiers destination existants ou lorsque les fichiers de destination n'existent pas.

Commande: cp

- La commande cp:
 - Copier un fichier : `cp fichier_source.txt dossier_destination/`
 - Copier un répertoire et son contenu : `cp -r repertoire_source/ dossier_destination/`
 - Copier plusieurs fichiers vers un répertoire de destination : `cp fichier1.txt fichier2.txt dossier_destination/`
 - Copier récursivement en affichant les fichiers copiés : `cp -rv repertoire_source/ dossier_destination/`
 - Copier un fichier en demandant confirmation avant de remplacer un fichier existant : `cp -i fichier_source.txt dossier_destination/`
 - Copier uniquement les fichiers source qui sont plus récents que les fichiers de destination existants : `cp -u fichier_source.txt dossier_destination/`

Commande: pwd et cat

- La commande pwd (print working directory) est utilisée pour afficher le répertoire de travail actuel dans un système d'exploitation de type Unix ou Linux.
- Voici comment l'utiliser : pwd
- La commande cat (concatenate) est utilisée pour afficher le contenu d'un ou plusieurs fichiers. Voici comment elle est utilisée :
 - cat [OPTIONS] [FICHIER(S)]
 - cat fichier.txt
- Options courantes :
 - -n : Numérote les lignes du fichier lors de l'affichage.
 - -b : Numérote les lignes non vides du fichier lors de l'affichage.
 - -E : Affiche un caractère \$ à la fin de chaque ligne.
 - -T : Affiche les caractères de tabulation comme ^I.
 - -v : Affiche les caractères non imprimables sous forme de codes ASCII.

Commande: nano

- La commande nano est un éditeur de texte en ligne de commande souvent utilisé dans les systèmes Unix et Linux. Il est conçu pour être simple et convivial, ce qui en fait un choix populaire pour les utilisateurs débutants et avancés.
- Syntaxe de base :
 - [OPTIONS] [FICHIER]
- Exemple d'utilisation :
 - nano fichier.txt
- Fichier .nanorc

Commande: mv

- La commande mv est utilisée pour déplacer ou renommer des fichiers et des répertoires dans un système Unix/Linux.
- Syntaxe de base :
 - mv [OPTIONS] SOURCE DESTINATION
- SOURCE : spécifie le chemin du fichier ou du répertoire source que vous souhaitez déplacer ou renommer.
- DESTINATION : spécifie le chemin où vous souhaitez déplacer le fichier ou le répertoire source, ou le nouveau nom si vous souhaitez le renommer.
- Exemples d'utilisation :
 - Déplacer un fichier vers un répertoire : mv fichier.txt dossier_destination/
 - Renommer un fichier : mv ancien_nom.txt nouveau_nom.txt
 - Déplacer un répertoire vers un autre répertoire : mv repertoire_source/ dossier_destination/
 - Renommer un répertoire : mv ancien_nom/ nouveau_nom/
- Options courantes :
 - -i, --interactive : Demande confirmation avant de remplacer un fichier existant.
 - -f, --force : Force le remplacement des fichiers existants sans confirmation.
 - -u, --update : Ne déplace que les fichiers source qui sont plus récents que les fichiers de destination existants ou lorsque les fichiers de destination n'existent pas.
 - -v, --verbose : Affiche des informations détaillées sur les fichiers déplacés.

Commande: rm

- La commande rm est utilisée pour supprimer des fichiers ou des répertoires dans un système Unix/Linux. Voici comment elle est utilisée.
- Syntaxe de base :
 - `rm [OPTIONS] FICHIER(S) OU REPERTOIRE(S)`
- FICHIER(S) OU REPERTOIRE(S) : spécifie le chemin des fichiers ou des répertoires que vous souhaitez supprimer.
- Exemples d'utilisation :
 - Supprimer un fichier : `rm fichier.txt`
 - Supprimer plusieurs fichiers en une seule commande : `rm fichier1.txt fichier2.txt fichier3.txt`
 - Supprimer un répertoire et son contenu récursivement : `rm -r repertoire_a_supprimer/`
- Options courantes :
 - `-i, --interactive` : Demande confirmation avant de supprimer chaque fichier ou répertoire.
 - `-f, --force` : Supprime les fichiers sans demander de confirmation, même s'ils sont en lecture seule ou si le fichier ou le répertoire n'existe pas.
 - `-r, -R, --recursive` : Supprime récursivement les répertoires et leur contenu.
 - `-v, --verbose` : Affiche des informations détaillées sur les fichiers supprimés.

Commande: Permissions

- Dans un système Unix/Linux, les fichiers et les répertoires sont associés à des permissions et à des propriétaires, ce qui contrôle l'accès aux fichiers et aux répertoires.
- Les permissions définissent ce qu'un utilisateur peut faire avec un fichier ou un répertoire. Il existe trois types de permissions :
 - Lecture (r) : Permet à l'utilisateur de lire le contenu du fichier ou du répertoire.
 - Écriture (w) : Permet à l'utilisateur de modifier le contenu du fichier ou du répertoire, d'y ajouter des fichiers dans le cas d'un répertoire, ou de le supprimer.
 - Exécution (x) : Pour un fichier, cela permet à l'utilisateur d'exécuter le fichier s'il s'agit d'un programme exécutable. Pour un répertoire, cela permet à l'utilisateur d'accéder à son contenu et de le parcourir.
- Les permissions sont définies pour trois types d'utilisateurs :
 - Propriétaire (u) : L'utilisateur qui a créé le fichier ou le répertoire.
 - Groupe (g) : Un groupe d'utilisateurs auquel le fichier ou le répertoire appartient.
 - Autres (o) : Tous les autres utilisateurs qui ne sont ni le propriétaire ni membres du groupe.

Commande: Permissions

- Le propriétaire d'un fichier ou d'un répertoire est l'utilisateur qui a créé le fichier ou le répertoire.
- Chaque fichier ou répertoire a un seul propriétaire, mais peut appartenir à un groupe.
- Pour afficher les permissions et les propriétaires d'un fichier ou d'un répertoire, vous pouvez utiliser la commande `ls -l` qui affiche une liste détaillée des fichiers et des répertoires avec leurs permissions et propriétaires.
- Pour modifier les permissions et les propriétaires d'un fichier ou d'un répertoire, vous pouvez utiliser les commandes **chmod** pour les permissions et `chown` pour les propriétaires.
- Exemple d'utilisation de `chmod` pour modifier les permissions d'un fichier :
 - `chmod u+rw fichier.txt`