**Commandes d’administration et réseau Shell**

**Assia EL GUERCH ID101**

1. **Administration :**

**Print Working Directory (PWD)**

* **Description :** Affiche le répertoire de travail actuel.
* **Commande :** pwd

**Change directory (CD)**

* **Description :** Change le répertoire de travail.
* **Commande :**
* cd /chemin/absolu
* cd chemin\_relatif

**List (LS)**

* **Description :** Liste les fichiers et les répertoires dans le répertoire actuel.
* **Commandes :**
* ls
* ls [repertoire]
* ls -a : affiche les fichiers cachés
* ls -l : affiche les détails
* ls -t : affiche par date
* ls -f : affiche par type
* ls -s : affiche par taille
* ls -r : affiche par ordre alphabétique
* la -x : affiche par extinction

**Make Directory (MKDIR)**

* **Description :** Crée un nouveau répertoire.
* **Commande :**
* mkdir dossier
* rmdir dossier : supprime un dossier vide
* mkdir dossier1 dossier2 dossier3 : crée plusieurs répertoires
* mkdir -p dossier1/dossier2/dossier3 : crée un chemin complet
* mkdir -p x1/x2/x3 y1/y2/y3 z1/z2/z3 : crée plusieurs chemins
* tree [dossier] : affiche l'arborescence des dossiers

**Create a file (Touch)**

* **Description :** Crée un nouveau fichier vide ou met à jour l'horodatage d'un fichier existant.
* **Commande :**
* touch fichier1
* touch f1 f2 f3 ou touch f{1,2,3}
* touch 'fichier numero un'

**Copier coller (CP)**

* **Description :** Copie des fichiers et des répertoires.
* **Commande :**
* cp [source] [destination]
* cp -r [source] [destination] : pour les répertoires

**Rename and Move (MV)**

* **Description :** Déplace ou renomme des fichiers et des répertoires.
* **Commande :**
* mv [source] [destination]
* mv old\_name new\_name
* mv -i : vérifier l’action [Y/N]
* mv -b : effectuer un sauvegard

**Remove (RM)**

* **Description :** Supprime des fichiers et des répertoires.
* **Commande :**
* rm [fichier]
* rm -r [repertoire]
* rm -i : pour confirmation
* rm -f : supprime forcément

**Editing (vi, cat)**

* **Description :** Édite ou affiche le contenu d'un fichier.
* **Commandes :**
* vi [fichier]
* cat [fichier]
* more [fichier]
* less [fichier]

**Gestion des utilisateurs (useradd, userdel, usermod)**

1. **useradd :** Ajoute un nouvel utilisateur au système.

* **Syntaxe :** useradd [options] [utilisateur]
* **Exemple :** useradd -m -s /bin/bash john
* **Options courantes :**
* -m : Crée le répertoire personnel de l'utilisateur.
* -s /chemin/vers/shell : Spécifie le shell par défaut de l'utilisateur.
* -u [UID] : Spécifie l'UID de l'utilisateur.
* -g [GID] : Spécifie le GID principal de l'utilisateur.
* -G [groupes] : Spécifie les groupes supplémentaires de l'utilisateur.
* -p : Spécifier un mot de passe
* -r : Donner le plus petit UID

1. **userdel :** Supprime un utilisateur du système.

* **Syntaxe :** userdel [options] [utilisateur]
* **Exemple :** userdel -r john
* **Options courantes :**
* -r : Supprime également le répertoire personnel de l'utilisateur.

1. **usermod :** Modifie les attributs d'un utilisateur existant.

* **Syntaxe :** usermod [options] [utilisateur]
* **Exemple :** usermod -aG groupe\_supplementaire john
* **Options courantes :**
* -aG [groupes] : Ajoute l'utilisateur aux groupes supplémentaires.
* -c [commentaire] : Ajoute un commentaire pour l'utilisateur.
* -d [répertoire] : Spécifie le répertoire personnel de l'utilisateur.
* -e [date] : Définit la date d'expiration du compte de l'utilisateur.
* -l [nouveau\_nom] : Renomme l'utilisateur.

**Gestionn des groupes (groupadd, groupmod, groupdel)**

1. **groupadd :** Crée un nouveau groupe sur le système.

* **Syntaxe** : groupadd [options] [nom\_groupe]
* **Exemple :** groupadd -g 1001 groupe1
* **Options courantes :**
* -g [GID] : Définit l'ID de groupe.

1. **groupmod :** Modifie les attributs d'un groupe existant.

* **Syntaxe :** groupmod [options] [nom\_groupe]
* **Exemple :** groupmod -n nouveau\_nom ancien\_nom
* **Options courantes :**
* -n [nouveau\_nom] : Renomme le groupe.
* **groupdel :** Supprime un groupe du système.
* **Syntaxe :** groupdel [options] [nom\_groupe]
* **Exemple :** groupdel groupe1

**Affichage des informations utilisateur et changer l’utilisateur**

* **id :** Affiche les informations sur un utilisateur, y compris son UID (User ID) et son GID (Group ID).
* **Syntaxe :** id [utilisateur]
* **Su [utilisateur] :** changer l’utilisateur
* **etc/logins.defs :** paramétres de cration des utilisateurs
* who: Affiche la liste des utilisateurs connectés à la station.
* finger [nom\_utilisateur]: Fournit des informations détaillées sur un utilisateur spécifique, y compris son nom, son identifiant, son répertoire personnel, son adresse électronique, et

**Affichage des utilisateurs et de leurs informations et mots de passe**

* **cat /etc/passwd :** Affiche la liste des utilisateurs et leurs informations associées.
* **cat /etc/shadow :** Affiche les mots de passe cryptés des utilisateurs.

**Modification du mot de passe d'un utilisateur**

* **passwd :** Permet à un utilisateur de modifier son propre mot de passe ou à l'administrateur de définir le mot de passe pour un autre utilisateur.
* **Syntaxe :** passwd [utilisateur]

**Gestion des droits (chown, chmod, chgrp, umask)**

1. **chown :** Change le propriétaire et/ou le groupe propriétaire d'un fichier ou d'un répertoire.

* **Syntaxe :** chown [utilisateur]:[groupe] [fichier]
* **Exemple :** chown john:users fichier.txt

1. **chmod :** Modifie les permissions d'accès d'un fichier ou d'un répertoire.

* **Syntaxe :** chmod [mode] [fichier]

Les permissions sont définies par des chiffres octaux ou par des symboles.

* **Exemples :**
* chmod 755 fichier.txt : Propriétaire peut lire, écrire et exécuter, autres peuvent seulement lire et exécuter.
* chmod u+x fichier.txt : Ajoute l'autorisation d'exécution au propriétaire.
* chmod go-r fichier.txt : Supprime les autorisations de lecture pour le groupe et les autres.

1. **umask :** Définit les permissions par défaut pour les nouveaux fichiers.

* Les permissions par défaut sont généralement définies dans le fichier de configuration /etc/profile ou /etc/bashrc.
* La valeur umask est soustraite des permissions maximales (777 pour les répertoires et 666 pour les fichiers) pour obtenir les permissions par défaut.
* **Exemple :**
* Si umask est défini à 002, les nouvelles permissions par défaut pour les fichiers seront 664 (666 - 002) et pour les répertoires seront 775 (777 - 002).

**Installation des packages**

* **yum :** Gestionnaire de paquets pour les distributions Linux basées sur RPM.
* yum -y install [packages] : Installe les paquets spécifiés avec confirmation automatique.
* yum -y remove [packages] : Supprime les paquets spécifiés avec confirmation automatique.
* yum -q : Vérifie les dépendances des paquets.
* **rpm :** Gestionnaire de paquets de bas niveau pour les distributions Linux basées sur RPM.
* rpm -qa : Liste tous les packages installés.

**Archiver et compresser**

* tar : Utilitaire d'archivage pour créer, extraire et manipuler des archives.
* tar -cvf [archive.tar] [dossier] : Crée un archive tar à partir du dossier spécifié.
* tar -zcvf [archive.tar.gz] [dossier] : Crée un archive tar et le compresse avec gzip.
* tar -zxvf [archive.tar.gz] : Décompresse et extrait une archive tar.gz.
* tar -jcvf [archive.tar.bz2] : Crée un archive tar et le compresse avec bzip2.
* tar -jxvf [archive.tar.bz2] : Décompresse et extrait une archive tar.bz2.
* **Options courantes :**
* c : Crée un nouvel archive.
* x : Extrait des fichiers d'une archive.
* t : Affiche le contenu d'une archive.
* r : Ajoute des fichiers à une archive.
* tar -tf [archive] : Affiche le contenu de l'archive sans l'extraire.
* tar -rvf [archive] [autre\_fichier] : Ajoute un autre fichier à une archive existante.
* tar -xvf [archive] : Extrait les fichiers d'une archive.
* **gzip : Utilitaire de compression de fichiers.**
* gzip [archive] : Compresse un fichier avec l'extension .gz.
* gunzip [archive.tar.gz] : Décompresse un fichier gzip.
* **bzip2 : Utilitaire de compression de fichiers.**
* bzip2 [archive] : Compresse un fichier avec l'extension .bz2.
* bunzip2 [archive.tar.bz2] : Décompresse un fichier bzip2.

**Commandes head et tail**

* head [fichier]: Affiche par défaut les 10 premières lignes du fichier spécifié.
* head -n x [fichier]: Affiche les x premières lignes du fichier.
* head -x [fichier]: Affiche x lignes du fichier.
* head -n 4 -v [fichier]: Affiche les 4 premières lignes du fichier avec le nom du fichier.
* head -n 5 -q [fichier]: Affiche les 5 premières lignes du fichier sans le nom du fichier.
* tail [fichier]: Affiche par défaut les 10 dernières lignes du fichier spécifié.
* tail -n x [fichier]: Affiche les x dernières lignes du fichier.
* tail -n 4 -v [fichier]: Affiche les 4 dernières lignes du fichier avec le nom du fichier.
* tail -n 5 -q [fichier]: Affiche les 5 dernières lignes du fichier sans le nom du fichier.
* tail -c [fichier]: Affiche les derniers caractères du fichier.
* head -n x [fichier] | tail -n x: Combinaison où tail applique ses résultats à la sortie de head.

**Trier les lignes avec sort**

* sort [fichier]: Trie les lignes du fichier par ordre alphabétique.
* sort -r [fichier]: Trie les lignes du fichier par ordre inverse.
* sort -n [fichier]: Trie les lignes du fichier comme des nombres, du plus petit au plus grand.
* sort -n -r [fichier]: Trie les lignes du fichier comme des nombres, du plus grand au plus petit.
* sort [fichier]: Trie les nombres comme des alphabets.
* sort -o test.txt test-tr.txt: Exporte les résultats du tri dans un fichier spécifié.
* sort -u [fichier]: Supprime les doublons lors du tri.

**Compter les mots avec wc**

* wc [fichier]: Affiche le nombre de lignes, de mots et de caractères du fichier.
* wc -l [fichier]: Affiche le nombre de lignes du fichier.
* wc -w [fichier]: Affiche le nombre de mots du fichier.
* wc -m [fichier]: Affiche le nombre de caractères du fichier.
* wc -c [fichier]: Affiche le nombre d'octets du fichier.

**Couper une partie du fichier avec cut**

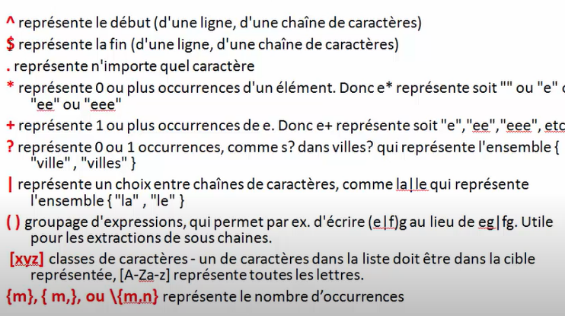
* cut -c x-x [fichier] ou cut -c x,x,x [fichier]: Sélectionne des caractères spécifiques ou des plages de caractères du fichier.
* cut -d [caractère] -f [numéro du champ] [fichier]: Spécifie le délimiteur dans le fichier et sélectionne le champ spécifié.

**Grep : Chercher dans un fichier**

* grep x [fichier]: Recherche le caractère x dans le fichier.
* grep -i x [fichier]: Recherche le caractère x dans le fichier sans faire de distinction entre majuscules et minuscules.
* grep -r x [dossier]: Recherche le caractère x dans tous les fichiers d'un dossier et de ses sous-dossiers.
* grep -v x [fichier]: Affiche les lignes du fichier qui ne contiennent pas le caractère x.
* grep -E x [fichier]: Utilise une expression régulière (regex) pour la recherche.
* Expressions régulières :
* ^: Début de ligne.
* .: Correspond à n'importe quel caractère.
* $: Fin de ligne.
* [ ]: Représente une plage de caractères permis.
* [^ ]: Représente une plage de caractères interdits.
* Exemples d'utilisation avec expressions régulières :
* grep '^pattern' [fichier]: Recherche les lignes commençant par 'pattern'.
* grep 'pattern$' [fichier]: Recherche les lignes se terminant par 'pattern'.
* grep '[abc]' [fichier]: Recherche les lignes contenant a, b ou c.
* grep '[^abc]' [fichier]: Recherche les lignes ne contenant pas a, b ou c.
* Caractères d'échappement :
* \: Utilisé pour échapper des caractères spéciaux comme '|', '{', '}'.
* Utilisation avec pipe et accolades OU :
* Utilisez le backslash \ pour échapper les caractères spéciaux comme '|', '{', '}' lorsque vous les utilisez avec grep.
* Caractères Jokers :
* . : Correspond à n'importe quel caractère unique.
* : Correspond à zéro ou plusieurs occurrences du caractère précédent.
* + : Correspond à une ou plusieurs occurrences du caractère précédent.
* ? : Correspond à zéro ou une seule occurrence du caractère précédent.
* {n} : Correspond à exactement n occurrences du caractère précédent.
* {n,} : Correspond à au moins n occurrences du caractère précédent.
* {n,m} : Correspond à entre n et m occurrences du caractère précédent.
* Exemples d'utilisation avec caractères jokers :
* grep 'abc.' [fichier]: Recherche les lignes contenant 'abc' suivi de n'importe quel caractère.
* grep 'a\*b' [fichier]: Recherche les lignes contenant zéro ou plusieurs 'a' suivis de 'b'.
* grep 'a+b' [fichier]: Recherche les lignes contenant une ou plusieurs 'a' suivis de 'b'.
* grep 'a?b' [fichier]: Recherche les lignes contenant zéro ou une 'a' suivie de 'b'.
* grep 'a{2}' [fichier]: Recherche les lignes contenant exactement deux 'a' consécutifs.
* grep 'a{2,}' [fichier]: Recherche les lignes contenant au moins deux 'a' consécutifs.
* grep 'a{1,3}b' [fichier]: Recherche les lignes contenant entre un et trois 'a' suivis de 'b'.

A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated

**Commande uniq : Supprimer les doublons dans un fichier**

* uniq [fichier]: Supprime les lignes en double consécutives dans le fichier.
* uniq [fichier] [nouveau\_fichier]: Écrit les résultats dans un nouveau fichier.
* uniq -c [fichier]: Compte combien de fois chaque ligne est répétée dans le fichier.
* uniq -d [fichier]: Affiche uniquement les lignes répétées (non uniques).

**Autres commandes :**

* date: Affiche la date et l'heure actuelles du système.
* cal: Affiche le calendrier du mois en cours.
* clear : reviens au début
* man : afficher le manual d’un commande
* diff [Ficher] [Ficher] : afficher les différences entre deux fichiers
* cmp [Ficher] [Ficher] : comparaison binaire de fichier
* join -i [fichier1] [fichier2]: Joint deux fichiers en fonction d'une colonne commune, en ignorant la casse des lettres.
* split -l [nombre\_lignes] [fichier]: Divise le contenu d'un fichier en plusieurs fichiers, chaque fichier contenant le nombre spécifié de lignes.
* tr -d [caractère]: Convertit ou supprime un caractère spécifié de l'entrée.

**Commande find : Rechercher des fichiers dans un répertoire**

* find -name [nom]: Recherche les fichiers par leur nom.
* find -print: Affiche les résultats de la recherche.
* find -type [type]: Recherche les fichiers par type (fichier régulier 'f', répertoire 'd', etc.).
* find -user [utilisateur] -group [groupe]: Recherche les fichiers appartenant à un utilisateur et à un groupe spécifiés.
* find -size [taille]: Recherche les fichiers par taille.
* find -atime [jours]: Recherche les fichiers accédés il y a [jours] jours.
* find -mtime [jours]: Recherche les fichiers modifiés il y a [jours] jours.
* find -ctime [jours]: Recherche les fichiers dont l'attribut de changement a été modifié il y a [jours] jours.
* find -perm [permissions]: Recherche les fichiers avec des autorisations spécifiques.
* find -ls: Liste les fichiers avec des détails comme le ls standard.

**Gestion des fichiers et des répertoires :**

* which [nom\_fichier ou répertoire]: Affiche le chemin absolu de l'exécutable spécifié.
* logname [nom\_utilisateur]: Affiche le nom de connexion actuel.
* startx: Démarre l'environnement de bureau X Window sous Linux.

**Gestion des utilisateurs et des sessions :**

* who: Affiche la liste des utilisateurs connectés à la station.
* logout [nom\_utilisateur]: Ferme la session de l'utilisateur spécifié.
* exit: Déconnecte l'utilisateur actuel.

**Gestion de l'arrêt et du redémarrage :**

reboot: Redémarre la machine.

* shutdown -r [nom\_utilisateur] ou shutdown -r: Arrête et redémarre la machine, optionnellement en spécifiant un utilisateur qui a les privilèges nécessaires.
* shutdown -h [nom\_utilisateur] ou shutdown -h: Arrête proprement la machine, optionnellement en spécifiant un utilisateur qui a les privilèges nécessaires.

**Statistiques et gestion des systèmes de fichiers :**

* df -h: Affiche l'utilisation de l'espace disque sur les systèmes de fichiers montés de manière lisible par l'homme (en gigaoctets, mégaoctets, etc.).
* df -i: Affiche l'utilisation des inodes sur les systèmes de fichiers montés.
* mkfs -t [type\_systeme\_de\_fichier] -V [options] [device]: Crée un système de fichier sur le périphérique spécifié avec le type de système de fichier donné et les options choisies.
* resize2fs [device] [taille]: Ajuste la taille d'un système de fichier ext2/ext3/ext4 à la taille spécifiée.
* fsck -A -N: Vérifie l'intégrité des systèmes de fichiers sans effectuer de réparation. Cette commande peut être utilisée pour prévisualiser les erreurs potentielles sans les corriger.

**Gestion des volumes physiques et logiques :**

* pvcreate [device]: Crée un volume physique sur le périphérique spécifié, préparant ainsi le périphérique pour une utilisation avec LVM (Logical Volume Manager).
* lvcreate [options] -n [nom\_volume] [groupe\_de\_volumes] [taille]: Crée un volume logique avec le nom spécifié et la taille donnée dans le groupe de volumes indiqué.

**Montage et démontage des systèmes de fichiers :**

* mount -a -t [type\_systeme\_de\_fichier]: Montre tous les systèmes de fichiers répertoriés dans /etc/fstab avec le type de système de fichier spécifié.
* umount -f -v [point\_de\_montage]: Démonte un système de fichier de manière forcée (force) et verbeuse (verbose), indiquant les actions effectuées.

**Autres commandes utiles :**

* fdisk -l: Liste les partitions de tous les disques disponibles sur la machine.

**Gestion des processus :**

* États d'un processus :
* R : Run (en cours d'exécution)
* S : Sleep
* D : Disk Sleep
* T : Stopped (Ctrl+z)
* Z : Zombie
* top: Affiche les processus en cours d'exécution avec des détails.
* top -u [nom\_utilisateur] ou top: Affiche les processus en cours d'exécution pour un seul utilisateur.
* jobs: Affiche les processus associés aux tâches (shell).
* ps -l: Affiche les processus de chaque utilisateur au moment de la connexion.
* ps -e: Affiche les processus du système d'exploitation.
* ps -ef: Affiche tous les processus avec tous les détails.
* ps -u [nom\_utilisateur]: Affiche les processus lancés par l'utilisateur actif.
* kill [signal] [PID]: Tue un processus avec le signal spécifié (15=terminer, 9=forcer, 19=suspendre, 18=reprendre).

**Gestion des services :**

* service [nom\_service] start: Démarre un service.
* service [nom\_service] stop: Arrête un service.
* service [nom\_service] restart: Redémarre un service.
* service [nom\_service] status: Affiche le statut d'un service.
* systemctl start [nom\_service]: Démarre un service (avec systemd).
* systemctl stop [nom\_service]: Arrête un service (avec systemd).
* systemctl restart [nom\_service]: Redémarre un service (avec systemd).
* systemctl status [nom\_service]: Affiche le statut d'un service (avec systemd).

1. **Reseaux et Sécurite**

**Pare-feu avec ufw (Uncomplicated Firewall) :**

* Configuration de base :
* ufw enable: Active le pare-feu.
* ufw disable: Désactive le pare-feu.
* ufw status: Affiche le statut du pare-feu et les règles actuelles.
* Gestion des règles :
* ufw allow [port/protocole]: Autorise le trafic entrant sur le port spécifié.
* ufw deny [port/protocole]: Bloque le trafic entrant sur le port spécifié.
* ufw delete [règle]: Supprime une règle spécifique.
* Configuration avancée :
* ufw default allow/deny: Définit la politique par défaut pour le trafic entrant.
* ufw reset: Réinitialise toutes les règles à leurs valeurs par défaut.
* ufw logging on/off: Active ou désactive les journaux de pare-feu.

**Pare-feu avec iptables :**

* Configuration de base :
* iptables -A INPUT -p [protocole] --dport [port] -j ACCEPT: Autorise le trafic entrant sur un port spécifié.
* iptables -A INPUT -s [adresse\_IP] -j DROP: Bloque le trafic entrant d'une adresse IP spécifiée.
* iptables -L: Liste toutes les règles du pare-feu.
* Gestion des règles :
* iptables -D [chaîne] [numéro\_de\_règle]: Supprime une règle spécifique.
* Configuration avancée :
* iptables-save: Enregistre les règles de pare-feu actuelles dans un fichier.
* iptables-restore: Restaure les règles de pare-feu à partir d'un fichier.

**Commandes réseau générales :**

* Configuration de l'interface réseau :
* ifconfig: Affiche les informations sur les interfaces réseau.
* ifconfig [interface] [adresse\_IP] netmask [masque\_sous-réseau]: Configure une adresse IP statique pour une interface.
* ifconfig [interface] up/down: Active ou désactive une interface.
* Diagnostic réseau :
* ping [adresse\_IP/URL]: Vérifie la connectivité avec une adresse IP ou une URL.
* traceroute [adresse\_IP/URL]: Affiche le chemin pris par les paquets vers une destination.
* netstat -tuln: Affiche les connexions réseau et les ports en écoute.
* Gestion des connexions :
* ssh [utilisateur]@[adresse\_IP]: Se connecte à un serveur distant via SSH.
* scp [fichier] [utilisateur]@[adresse\_IP]:[destination]: Copie des fichiers de/vers un serveur distant via SSH.
* wget [URL]: Télécharge un fichier depuis une URL.
* Configuration DNS :
* nslookup [URL]: Effectue une requête DNS pour obtenir des informations sur une URL.
* dig [URL]: Effectue une requête DNS avancée

**Commandes pour tester la connectivité et surveiller les connexions réseau :**

* ping [adresse\_IP/URL]: Envoie des paquets ICMP à une adresse IP ou une URL pour tester la connectivité et vérifier si l'hôte distant est accessible.
* netstat: Affiche les connexions réseau actives, les tables de routage, les statistiques d'interface et d'autres informations sur les connexions réseau.
* ifconfig: Affiche les informations sur les interfaces réseau de la machine, telles que les adresses IP, les masques de sous-réseau et les informations d'état.
* traceroute [adresse\_IP/URL]: Affiche le chemin pris par les paquets IP entre la machine locale et une adresse IP ou une URL distante, en montrant les sauts (hops) intermédiaires.
* dig [URL] : Effectue une requête DNS pour obtenir des informations détaillées sur un nom de domaine, telles que les enregistrements de ressources (A, CNAME, MX, etc.).
* host [URL] : Effectue des requêtes DNS pour résoudre un nom de domaine en adresse IP et vice versa.
* curl [URL] : Télécharge un fichier ou une ressource depuis une URL spécifiée via HTTP, HTTPS, FTP ou d'autres protocoles supportés.
* whois [nom\_de\_domaine] : Interroge la base de données WHOIS pour obtenir des informations sur le propriétaire d'un nom de domaine, l'enregistrement du domaine, etc.
* ifplugstatus: Indique si un câble réseau est connecté ou déconnecté à une interface réseau.

1. **Creation d’un script Bash**

**Structure de base d'un script Bash :**

#!/bin/bash

# Votre code commence ici

* La première ligne #!/bin/bash indique au système d'exploitation qu'il s'agit d'un script Bash et où se trouve l'interpréteur Bash.
* Les lignes de commentaire commençant par # sont ignorées par l'interpréteur et servent à documenter votre script.

**Variables :**

* variable=value : Définit une variable.
* variable= : Représente une valeur nulle ou non définie.
* Utilisez les variables en les préfixant avec $. Par exemple : echo "Bonjour, $nom".

**Entrées et sorties :**

* read variable : Lit une entrée de l'utilisateur dans la console et la stocke dans une variable.
* echo "texte" : Affiche du texte dans la console.
* echo "texte" ou printf "texte" : Envoie du texte à la sortie standard.

**Structures de contrôle :**

* **if, elif, else : Exécute des blocs de code conditionnellement.**

if [ condition ]; then

# Bloc de code si la condition est vraie

elif [ autre\_condition ]; then

# Bloc de code si la première condition est fausse, mais cette condition est vraie

else

# Bloc de code si aucune des conditions précédentes n'est vraie

fi

* **case/esac : Exécute des blocs de code en fonction de la valeur d'une expression.**

case "$variable" in

valeur1) # Bloc de code si la valeur est "valeur1" ;;

valeur2) # Bloc de code si la valeur est "valeur2" ;;

\*) # Bloc de code par défaut ;;

esac

**Boucles :**

* **for : Exécute un bloc de code un nombre spécifié de fois.**

for ((i = 1; i <= 5; i++)); do

# Bloc de code à répéter

done

* **for : Parcourt les éléments d'une collection et exécute un bloc de code pour chaque élément.**

for element in "${tableau[@]}"; do

# Bloc de code à exécuter pour chaque élément

done

* **for : Boucle à travers une liste d'éléments.**

for element in liste

do

# Faire quelque chose avec chaque élément

done

* **while : Exécute un bloc de code tant qu'une condition spécifiée est vraie.**

while [ condition ]; do

# Bloc de code à répéter tant que la condition est vraie

done

**Tableaux :**

* tableau=(élément1 élément2 élément3) : Définit un tableau.
* ${tableau[index]} : Accède à un élément spécifique du tableau.
* for element in "${tableau[@]}"; do ... done : Parcourt tous les éléments du tableau.
* ${#tableau[@]} : Obtient le nombre d'éléments dans le tableau.

**Opérations logiques :**

* && : Opérateur logique ET.
* || : Opérateur logique OU.
* ! : Opérateur logique NON.

**Opérations de comparaison :**

* == : Égal à.
* != : Différent de.
* -gt : Supérieur à.
* -lt : Inférieur à.
* -ge : Supérieur ou égal à.
* -le : Inférieur ou égal à.

**Opérations mathématiques :**

* + : Addition.
* - : Soustraction.
* : Multiplication.
* / : Division.
* % : Modulo (reste de la division).