# 1LADM -TP 401 – Gestion des processus et des tâches

### **Exercice sur la Commande chmod**

#### Objectifs:

* Comprendre comment modifier les permissions des fichiers pour sécuriser le contenu et permettre des utilisations spécifiques.

#### Introduction:

**chmod** est une commande utilisée pour changer les modes d'accès d'un fichier ou répertoire. Cela inclut les permissions de lecture, d'écriture et d'exécution accordées au propriétaire du fichier, au groupe auquel appartient le fichier, et aux autres utilisateurs.

#### Questions de Réflexion:

1. Pourquoi est-il important de pouvoir changer les permissions d'un fichier sous Linux ?

Réponses:

1. Comment les permissions incorrectes sur un fichier exécutable pourraient-elles affecter la sécurité d'un système ?

Réponses:

1. Quels seraient les risques de mettre les permissions de tous les fichiers à **777** ?

Réponses:

#### Exercice Pratique:

**Partie A: Exploration des Permissions**

1. Ouvrez un terminal sur votre système Linux.
2. Créez un fichier test avec **touch testfile**.
3. Utilisez la commande **ls -l testfile** pour voir les permissions actuelles du fichier.

Coller le résultat des permissions ici et expliquer les différentes permissions attribuées à ce fichier:

1. Changez les permissions du fichier pour que seul le propriétaire puisse lire et écrire, en utilisant **chmod 600 testfile**.

Fournir le résultat ici:

1. Vérifiez les nouvelles permissions avec **ls -l testfile**.

Afficher le résultat:

**Partie B: Questions spécifiques**

1. Changez les permissions du fichier **testfile** pour permettre à tous les utilisateurs de lire le fichier mais à personne de l'écrire ou de l'exécuter. Quelle commande utiliseriez-vous ?

Réponses:

1. Expliquez l'effet de la commande **chmod +x testfile**. Qui peut exécuter le fichier après cette commande ?

Réponses:

1. Dans quel scénario seriez-vous amené à utiliser **chmod 755** sur un dossier ?

Réponse:

#### Pour aller plus loin:

* Modifiez les permissions d'un script que vous avez écrit pour permettre à d'autres utilisateurs de votre groupe de l'exécuter, mais pas de le modifier. Notez la commande utilisée et les raisons de ce choix. (Effectuez cette tâche uniquement sous la supervision d'un enseignant ou sur une machine virtuelle pour éviter des modifications non souhaitées sur votre système.)

### **Exercice sur la Commande chown**

#### Objectifs:

* Comprendre comment changer le propriétaire d'un fichier ou d'un dossier et pourquoi cela peut être nécessaire dans un environnement Linux.

#### Introduction:

La commande **chown** (change owner) est utilisée pour modifier le propriétaire et/ou le groupe d'un fichier ou répertoire. Cela est crucial pour la gestion des droits d'accès et la sécurité des systèmes de fichiers.

#### Questions de Réflexion:

1. Pourquoi un administrateur système pourrait-il avoir besoin de changer le propriétaire d'un fichier ?

Réponses:

1. Quels problèmes de sécurité peuvent survenir si un fichier sensible est possédé par un utilisateur non privilégié ?

Réponses:

1. Comment l'utilisation de **chown** en conjonction avec **chmod** peut-elle améliorer la sécurité d'un fichier ?

Réponses:

#### Exercice Pratique:

**Partie A: Exploration de chown**

1. Ouvrez un terminal sur votre système Linux.
2. Créez un fichier nommé **example.txt** et un dossier nommé **example\_dir** avec la commande **touch example.txt** et **mkdir example\_dir**.
3. Affichez les propriétaires actuels avec **ls -l example.txt example\_dir**.

Coller le résultat ici:

1. Changez le propriétaire de **example.txt** à un utilisateur (remplacez **newuser** par un utilisateur existant sur votre système) avec **sudo chown newuser example.txt**.

Afficher le résultat ici:

1. Vérifiez que le changement a été effectué avec **ls -l example.txt**.

Afficher le résultat ici:

**Partie B: Questions spécifiques**

1. Si vous avez un dossier avec plusieurs fichiers et sous-dossiers, comment changeriez-vous le propriétaire de tous les contenus à **newuser** ?

Réponses:

1. Quelle commande utiliseriez-vous pour changer à la fois le propriétaire et le groupe d'un fichier ?

Réponses:

1. Que se passe-t-il si vous essayez de changer le propriétaire d'un fichier sans les permissions appropriées ?

Réponses:

#### Pour aller plus loin:

* Essayez de changer le propriétaire de plusieurs fichiers en utilisant un seul appel de commande. Pensez à un scénario où vous auriez besoin de réassigner la propriété de fichiers lors d'un changement d'un membre de l'équipe. Documentez les commandes utilisées et les raisons derrière votre choix.

### **Exercice sur la Commande chgrp**

#### Objectifs:

* Apprendre à changer le groupe associé à des fichiers ou des répertoires et comprendre pourquoi cela est crucial pour la gestion des permissions d'accès sur un système Linux.

#### Introduction:

La commande **chgrp** (change group) est utilisée pour changer le groupe d'un ou plusieurs fichiers ou répertoires. Cette commande est essentielle pour gérer efficacement les permissions de groupe sur un système de fichiers, facilitant ainsi la collaboration entre différents utilisateurs.

#### Questions de Réflexion:

1. Quand serait-il nécessaire de changer le groupe associé à un fichier ou à un répertoire?

Réponses:

1. Comment la modification des groupes d'un fichier peut-elle affecter l'accès à ce fichier ?

Réponses:

1. Quelles sont les différences et les implications entre changer un groupe avec **chgrp** et modifier les permissions avec **chmod** ?

Réponses:

#### Exercice Pratique:

**Partie A: Exploration de chgrp**

1. Ouvrez un terminal sur votre système Linux.
2. Créez un fichier et un répertoire de test avec **touch groupfile.txt** et **mkdir groupdir**.
3. Changez le groupe de **groupfile.txt** au groupe **staff** (assurez-vous que ce groupe existe, ou remplacez-le par un groupe valide sur votre système) en utilisant **sudo chgrp staff groupfile.txt**.
4. Vérifiez le changement avec **ls -l groupfile.txt**.

Afficher le résultat ici:

**Partie B: Questions spécifiques**

1. Comment changeriez-vous le groupe de tous les fichiers et sous-répertoires dans **groupdir** pour qu'ils appartiennent au groupe **staff** ?

Réponses et si possible faire le test et afficher le résultat:

1. Quelle commande utiliseriez-vous pour voir tous les groupes auxquels un utilisateur spécifique appartient, et comment cela pourrait influencer votre décision de changer de groupe pour un fichier?

Réponses:

1. Discutez de l'importance de choisir correctement le groupe pour des fichiers partagés dans un environnement de travail collaboratif.

Réponses:

#### Pour aller plus loin:

* Imaginez que vous devez préparer un répertoire pour une collaboration entre plusieurs départements. Créez un répertoire, attribuez-lui des fichiers test, et configurez les groupes pour refléter les différents départements qui doivent accéder aux fichiers. Documentez les commandes que vous utilisez et expliquez pourquoi vous avez choisi ces groupes spécifiques.

### **Exercice sur la Commande free**

#### Objectifs:

* Comprendre l'utilisation de la mémoire et de la swap sur un système Linux en utilisant la commande **free**.

#### Introduction:

La commande **free** est utilisée pour afficher la quantité de mémoire libre et utilisée sur le système Linux. Elle donne des informations détaillées sur la RAM physique et l'espace de swap, ce qui est essentiel pour surveiller et gérer les ressources système efficacement.

#### Questions de Réflexion:

1. Pourquoi est-il important de surveiller l'utilisation de la mémoire dans un système Linux?

Réponses:

1. Qu'est-ce que la mémoire swap et dans quelles circonstances est-elle utilisée?

Réponses:

1. Comment l'utilisation excessive de la swap peut-elle affecter les performances du système?

Réponses:

#### Exercice Pratique:

**Partie A: Exploration de free**

1. Ouvrez un terminal sur votre système Linux.
2. Exécutez la commande **free -h** pour afficher les informations de mémoire dans un format lisible par l'homme (c'est-à-dire en GiB ou MiB).

Afficher le résultat ici :

1. Notez la mémoire totale, utilisée, libre, et la mémoire swap disponible.

Réponses:

**Partie B: Questions spécifiques**

1. Comment interpréteriez-vous une situation où la majorité de la swap est utilisée?

Réponses :

1. Si vous constatez que votre système utilise régulièrement une grande quantité de swap, quelle stratégie pourriez-vous envisager pour améliorer les performances?

Réponses:

1. Évaluez l'impact potentiel de l'ajout de RAM physique supplémentaire par rapport à l'optimisation des applications pour utiliser moins de mémoire. Quelles seraient les implications pour les performances globales?

Réponses:

#### Pour aller plus loin:

* Imaginez que vous gérez un serveur qui commence à ralentir pendant les périodes de forte charge. Utilisez la commande **free** régulièrement pour surveiller l'utilisation de la mémoire sur une semaine. Enregistrez les résultats et analysez les tendances de l'utilisation de la mémoire et de la swap. Proposez des modifications ou des améliorations basées sur vos observations pour optimiser les performances du serveur.

### **Exercice sur la Commande ps**

#### Objectifs:

* Comprendre comment utiliser la commande **ps** pour visualiser les processus actifs sur un système Linux et apprendre à interpréter les informations qu'elle fournit.

#### Introduction:

La commande **ps** (process status) est un outil puissant sous Linux qui permet d'afficher des informations sur les processus en cours d'exécution. Elle est essentielle pour le monitoring et la gestion des processus système.

#### Questions de Réflexion:

1. Pourquoi est-il utile de connaître les processus en cours sur un système Linux?

Réponses:

1. Comment les informations fournies par **ps** peuvent-elles aider à résoudre des problèmes système?

Réponses:

1. Quelles sont les implications de tuer un processus sans comprendre son rôle ?

Réponses:

#### Exercice Pratique:

**Partie A: Exploration de ps**

1. Ouvrez un terminal sur votre système Linux.
2. Exécutez la commande **ps aux** pour afficher tous les processus en cours sur le système avec une liste détaillée.
3. Identifiez un processus lié à une application spécifique que vous avez ouverte, comme un navigateur ou un éditeur de texte.

Afficher le résultat :

**Partie B: Questions spécifiques**

1. Comment filtreriez-vous la sortie de **ps** pour ne montrer que les processus appartenant à un utilisateur spécifique ?

Afficher la commande + résultat :

1. Expliquez l'importance de la colonne **PID** et comment elle est utilisée avec d'autres commandes système.

Réponses:

1. Que signifient les %CPU et %MEM dans la sortie de **ps** et comment ces valeurs affectent-elles les performances du système?

Réponses;

#### Pour aller plus loin:

* Exécutez une application gourmande en ressources, puis utilisez **ps** pour observer comment elle affecte l'utilisation des ressources système. Tentez d'identifier des options pour **ps** qui vous permettraient de trier les processus par utilisation de la mémoire ou du CPU.

### **Exercice sur la Commande kill / killall**

#### Objectifs:

* Apprendre à utiliser les commandes **kill** et **killall** pour gérer efficacement les processus sous Linux en terminant les processus indésirables ou bloqués.

#### Introduction:

Les commandes **kill** et **killall** sont utilisées pour envoyer des signaux aux processus. Alors que **kill** envoie un signal à un processus spécifique identifié par son PID (Process ID), **killall** envoie un signal à tous les processus portant le nom spécifié.

#### Questions de Réflexion:

1. Pourquoi est-il nécessaire d'avoir la capacité de terminer des processus sous Linux ?
2. Quels risques sont associés à l'utilisation incorrecte des commandes **kill** et **killall** ?
3. Comment pouvez-vous vous assurer que vous terminez un processus de manière sécuritaire ?

#### Exercice Pratique:

**Partie A: Exploration de kill**

1. Ouvrez un terminal sur votre système Linux.
2. Lancez une application que vous pouvez fermer sans risque, comme un éditeur de texte ou un terminal supplémentaire.
3. Utilisez **ps aux | grep [nom de l'application]** pour trouver le PID de l'application.
4. Utilisez **kill [PID]** pour terminer le processus. Si le processus ne se termine pas avec le signal par défaut, essayez **kill -9 [PID]**.

**Partie B: Exploration de killall**

1. Ouvrez plusieurs instances d'une application (par exemple, **gedit**).
2. Utilisez **killall gedit** pour fermer toutes les instances de cette application.
3. Vérifiez si toutes les instances ont été fermées.

**Questions spécifiques**

1. Quelle commande utiliseriez-vous pour envoyer un signal SIGINT à un processus ?
2. Que se passe-t-il si vous exécutez **killall** sur un processus qui n'existe pas ?
3. Comment **kill -9** est-il différent des autres formes de la commande **kill** ?

#### Pour aller plus loin:

* Créez un script qui lance plusieurs instances d'un processus, surveille leur utilisation des ressources, et termine ceux qui dépassent un certain seuil de CPU ou de mémoire. Utilisez **ps** pour le monitoring et **kill** ou **killall** pour la gestion des processus.

### **Exercice sur la Commande shutdown**

#### Objectifs:

* Maîtriser l'utilisation de la commande **shutdown** pour arrêter ou redémarrer un système Linux de manière contrôlée et sécurisée.

#### Introduction:

La commande **shutdown** est utilisée pour arrêter ou redémarrer un système Linux. Elle permet de terminer les processus en cours d'exécution de manière propre, en évitant les pertes de données ou la corruption du système de fichiers, ce qui peut se produire lors d'un arrêt forcé.

#### Questions de Réflexion:

1. Pourquoi est-il préférable d'utiliser la commande **shutdown** plutôt que de simplement éteindre l'alimentation d'un ordinateur ?

Réponses:

1. Quelles sont les différences entre un arrêt (shutdown), un redémarrage (reboot), et une mise en veille (suspend) ?

Réponses:

1. Quelles commandes supplémentaires pourraient être utiles pour gérer un système avant de l'arrêter complètement ?

Réponses:

#### Exercice Pratique:

**Partie A: Exploration de shutdown**

1. Ouvrez un terminal sur votre système Linux (assurez-vous de ne pas fermer de système de production ou de système critique).
2. Planifiez un arrêt du système dans 5 minutes pour permettre à tous les processus de terminer proprement en utilisant :
3. **sudo shutdown -P +5**
4. Expliquez que l'option **-P** signifie "power off" après l'arrêt et **+5** indique le délai en minutes.

Réponses:

1. Annulez l'arrêt planifié si nécessaire avec :

**sudo shutdown -c**

1. Discutez de pourquoi il pourrait être nécessaire d'annuler un arrêt.

Réponses:

**Partie B: Exploration du redémarrage**

1. Planifiez un redémarrage dans 10 minutes :
2. **sudo shutdown -r +10 "System will reboot for maintenance."**

Le **-r** signifie "reboot" et le message après la commande informe les utilisateurs connectés de la raison du redémarrage.

**Questions spécifiques**

1. Que feriez-vous si vous devez arrêter immédiatement le système pour une maintenance d'urgence ? Quelle commande utiliseriez-vous ?
2. Comment la commande **shutdown** affecte-t-elle les utilisateurs actuellement connectés au système ?
3. Expliquez l'importance de la planification de l'arrêt des serveurs dans un environnement d'entreprise.

#### Pour aller plus loin:

* Simulez la gestion de l'arrêt et du redémarrage de plusieurs serveurs Linux dans un environnement virtuel. Créez des scénarios où vous devez communiquer avec des équipes, planifier des arrêts pour des fenêtres de maintenance, et minimiser l'impact sur les opérations de l'entreprise.

### **Exercice sur la Commande adduser**

#### Objectifs:

* Apprendre à utiliser la commande **adduser** pour créer des comptes utilisateurs sur un système Linux, en respectant les bonnes pratiques pour la sécurité et la gestion des utilisateurs.

#### Introduction:

La commande **adduser** est un utilitaire sous Linux qui facilite la création de nouveaux comptes utilisateurs. Elle permet aux administrateurs de systèmes de configurer rapidement de nouveaux utilisateurs avec les bons paramètres de répertoire home, de groupe utilisateur, et de shell par défaut.

#### Questions de Réflexion:

1. Pourquoi est-il préférable d'utiliser **adduser** plutôt que **useradd**, une autre commande couramment utilisée pour créer des utilisateurs sous Linux?

Réponse:

1. Quelles informations sont nécessaires et quelles options pouvez-vous configurer lors de la création d'un utilisateur avec **adduser**?

Réponse:

1. Comment la création appropriée d'utilisateurs peut-elle influencer la sécurité globale du système?

Réponse

#### Exercice Pratique:

**Partie A: Création d'un utilisateur**

1. Ouvrez un terminal sur votre système Linux.
2. Utilisez la commande **sudo adduser newuser**, en remplaçant **newuser** par le nom de l'utilisateur que vous souhaitez créer. Suivez les instructions à l'écran pour définir le mot de passe et remplir les informations facultatives de l'utilisateur.
3. Une fois l'utilisateur créé, vérifiez que le répertoire home a été créé et que les fichiers de configuration par défaut y sont présents en naviguant vers **/home/newuser** et en listant le contenu avec **ls -la**.

**Partie B: Questions spécifiques**

1. Quelle commande utiliseriez-vous pour voir tous les utilisateurs actuellement définis sur le système Linux?
2. Comment modifieriez-vous les paramètres par défaut pour la commande **adduser** (comme le shell par défaut ou le répertoire home)?
3. Quels fichiers de configuration système **adduser** modifie-t-elle lors de la création d'un nouveau compte utilisateur?

#### Pour aller plus loin:

* Envisagez de créer un script qui automatise la création d'utilisateurs en fonction d'une liste fournie dans un fichier texte. Chaque utilisateur pourrait avoir des paramètres personnalisés basés sur les données du fichier. Documentez les commandes utilisées et expliquez les choix de configuration pour chaque utilisateur.

### **Exercice sur la Commande passwd**

#### Objectifs:

* Apprendre à utiliser la commande **passwd** pour changer les mots de passe des utilisateurs sur un système Linux, garantissant ainsi la sécurité et la confidentialité des comptes utilisateur.

#### Introduction:

La commande **passwd** est essentielle pour la gestion des mots de passe des utilisateurs sous Linux. Elle permet non seulement aux utilisateurs de changer leur propre mot de passe, mais aussi aux administrateurs de modifier les mots de passe des autres utilisateurs pour maintenir la sécurité des systèmes.

#### Questions de Réflexion:

1. Pourquoi est-il important que les mots de passe soient changés régulièrement?

Réponses:

1. Quels risques un système pourrait-il encourir si les mots de passe des utilisateurs ne sont pas suffisamment sécurisés?

Réponses:

1. Comment la commande **passwd** peut-elle contribuer à prévenir l'accès non autorisé à un système?

Réponses:

#### Exercice Pratique:

**Partie A: Changement de mot de passe**

1. Ouvrez un terminal sur votre système Linux.
2. Si vous avez les droits d'administrateur, changez le mot de passe d'un utilisateur test (créez un nouvel utilisateur si nécessaire avec **sudo adduser testuser**) en utilisant :

.

**Partie B: Questions spécifiques**

1. Quelles mesures pouvez-vous prendre pour garantir que les mots de passe choisis sont forts et sécurisés?

Réponses:

1. Comment un administrateur système peut-il forcer tous les utilisateurs à changer leur mot de passe lors de leur prochaine connexion?

Réponses + commandes associées

#### Pour aller plus loin:

* Créez un scénario de gestion des mots de passe pour une petite entreprise, incluant des politiques de mot de passe, des procédures de changement régulier, et des audits de sécurité. Utilisez **passwd** et d'autres outils Linux pour illustrer comment vous implémenteriez ces politiques.

### **Exercice sur la Commande usermod**

#### Objectifs:

* Utiliser **usermod** pour modifier les comptes utilisateurs existants et comprendre les implications de ces modifications dans la gestion des utilisateurs et de la sécurité.

#### Introduction:

**usermod** est un outil pour modifier un compte utilisateur existant. Les modifications peuvent inclure le changement du nom d'utilisateur, du répertoire home, du shell par défaut, ou des groupes auxquels l'utilisateur appartient.

#### Questions de Réflexion:

1. Quand serait-il nécessaire de modifier un compte utilisateur avec **usermod** ?

Réponses:

1. Quelles sont les implications de changer le répertoire home d'un utilisateur ?

Réponses:

1. Comment les modifications apportées par **usermod** peuvent-elles affecter les processus en cours d'exécution ou les services gérés par l'utilisateur modifié ?

Réponses:

#### Exercice Pratique:

**Partie A: Modification d'un utilisateur**

1. Ouvrez un terminal sur votre système Linux.
2. Créez un utilisateur de test **testuser** avec **sudo adduser testuser**.
3. Modifiez le shell par défaut de l'utilisateur pour **/bin/zsh** avec **sudo usermod -s /bin/zsh testuser**. Vérifiez le changement en examinant le fichier **/etc/passwd**.

Afficher le résultat ici :

1. Changez le répertoire home de l'utilisateur en **/newhome/testuser** avec **sudo usermod -d /newhome/testuser -m testuser** et vérifiez que les fichiers ont été déplacés correctement.

**Partie B: Questions spécifiques**

1. Comment **usermod** peut-il être utilisé pour ajouter un utilisateur à plusieurs groupes simultanément ?

Réponses:

1. Quelles commandes supplémentaires pourraient être nécessaires pour s'assurer que les modifications n'affectent pas négativement les services système ?

Réponses:

1. Discutez des considérations de sécurité lors de l'utilisation de **usermod** pour changer les groupes d'un utilisateur dans un environnement multi-utilisateur.

Réponses:

#### Pour aller plus loin:

* Simulez un scénario où vous devez réorganiser les répertoires home de plusieurs utilisateurs à la suite d'une modification de la politique de sécurité. Planifiez et exécutez ces changements en utilisant **usermod**, en prenant en compte les permissions, les processus en cours et les sauvegardes nécessaires.

### **Exercice sur la Commande apt-get / apt**

#### Objectifs:

* Maîtriser l'utilisation des commandes **apt-get** et **apt** pour la gestion des paquets sur les systèmes Debian et Ubuntu, en se concentrant sur l'installation, la mise à jour, et la suppression de logiciels.

#### Introduction:

**apt-get** et **apt** sont des outils pour la gestion des paquets dans les distributions basées sur Debian, permettant aux utilisateurs de gérer facilement les logiciels installés sur leurs systèmes à travers une interface de ligne de commande.

#### Questions de Réflexion:

1. Quelles sont les différences entre **apt-get** et **apt**, et pourquoi **apt** est-il souvent préféré dans les interactions courantes ?

Réponses:

1. Comment la gestion correcte des paquets peut-elle affecter la performance et la sécurité d'un système ?

Réponses:

1. Quels problèmes pourriez-vous rencontrer lors de l'installation ou de la mise à jour de paquets et comment les résoudre ?

Réponses:

#### Exercice Pratique:

**Partie A: Gestion des paquets**

1. Ouvrez un terminal sur votre système Linux.
2. Mettez à jour la liste des paquets disponibles avec **sudo apt update**.
3. Installez un paquet, par exemple **prometheus**, avec **sudo apt install prometheus**.
4. Vérifiez si des mises à jour de paquets sont disponibles avec **sudo apt list --upgradable**.
5. Mettez à jour les paquets installés avec **sudo apt upgrade**.
6. Supprimez le paquet installé avec **sudo apt remove tree** et nettoyez les paquets non nécessaires avec **sudo apt autoremove**.

**Partie B: Questions spécifiques**

1. Comment listeriez-vous tous les paquets installés sur votre système?

Réponses:

1. Quelle commande utiliseriez-vous pour rechercher un paquet spécifique disponible dans les dépôts?

Réponses:

1. Expliquez comment **apt** gère les dépendances lors de l'installation ou de la suppression de paquets.

Réponses:

#### Pour aller plus loin:

* Simulez une situation où vous devez installer, configurer, et ensuite purger un logiciel, en s'assurant que toutes ses dépendances et configurations sont également nettoyées. Documentez le process

### **Exercice sur la Commande systemctl**

#### Objectifs:

* Comprendre l'utilisation de **systemctl** pour gérer les services et les unités systemd dans un système Linux, en mettant l'accent sur le contrôle des services système.

#### Introduction:

**systemctl** est un outil essentiel pour gérer le système et les services sous Linux utilisant systemd, le système d'init et de gestion de systèmes le plus couramment utilisé dans les distributions modernes de Linux. Il permet de démarrer, arrêter, redémarrer, recharger et obtenir le statut des services, ainsi que d'autres actions.

#### Questions de Réflexion:

1. Pourquoi est-il crucial de pouvoir gérer les services système avec **systemctl**?

Réponses;

1. Comment **systemctl** aide-t-il à maintenir la stabilité et la sécurité d'un système Linux?

Réponses:

1. Quelles pourraient être les conséquences de l'arrêt ou du redémarrage incorrect d'un service critique?

Réponses:

#### Exercice Pratique:

**Partie A: Gestion de services**

1. Ouvrez un terminal sur votre système Linux.
2. Utilisez **systemctl status prometheus** pour vérifier l'état du service SSH.

Afficher le résultat et le commenter (PID, commande exécutée, chemin du services etc ..) :

1. Essayez de démarrer, arrêter, et redémarrer le service prometheus en utilisant respectivement **sudo systemctl start prometheus.service**, **sudo systemctl stop prometheus.service**, et **sudo systemctl restart prometheus.service**.
2. Activez un service pour qu'il démarre au boot en utilisant **sudo systemctl enable prometheus.service** et désactivez-le avec **sudo systemctl disable prometheus.service**.

Pour quoi est il important d’activer vos services ?

Réponses :

**Partie B: Questions spécifiques**

1. Expliquez l'importance de la commande **systemctl daemon-reload** et dans quelles situations elle est nécessaire.

Réponses :

#### Pour aller plus loin:

* Configurez un nouveau service systemd pour une application fictive. Créez un fichier de service, placez-le dans le répertoire approprié, et utilisez **systemctl** pour le gérer. Documentez chaque étape et expliquez pourquoi chaque configuration a été choisie.

### **Exercice sur la Commande mount**

#### Objectifs:

* Comprendre comment monter des systèmes de fichiers sur un système Linux et l'importance de cette opération pour l'accès aux données.

#### Introduction:

La commande **mount** est utilisée pour monter des systèmes de fichiers sur des points de montage spécifiques dans la structure de répertoire d'un système Linux. Cette commande est cruciale pour accéder aux disques durs, aux disques USB, aux partages de réseau et à d'autres types de supports de stockage.

#### Questions de Réflexion:

1. Pourquoi le montage de différents types de systèmes de fichiers est-il essentiel dans la gestion des systèmes Linux?
2. Comment les permissions et les options de montage affectent-elles l'accès et la sécurité des données sur les systèmes de fichiers montés?
3. Quels problèmes pourraient survenir si un système de fichiers n'est pas correctement monté ou si les options de montage ne sont pas correctement configurées?

#### Exercice Pratique:

**Partie A: Exploration de mount**

1. Ouvrez un terminal sur votre système Linux.
2. Insérez une clé USB (assurez-vous qu'elle est connectée mais pas encore montée).
3. Utilisez la commande **lsblk** pour identifier le périphérique de la clé USB.
4. Montez la clé USB sur un point de montage existant, tel que **/mnt** (vous devrez peut-être créer un sous-dossier, par exemple avec **sudo mkdir /mnt/usb**).
5. Utilisez **mount /dev/sdx1 /mnt/usb** pour monter la clé USB, en remplaçant **/dev/sdx1** par le nom correct de votre périphérique.
6. Vérifiez que le montage a été effectué avec succès en utilisant **ls /mnt/usb** et **df -h**.

**Partie B: Questions spécifiques**

1. Comment vérifieriez-vous quels systèmes de fichiers sont actuellement montés sur votre système?
2. Si vous vouliez que la clé USB soit montée automatiquement à chaque démarrage du système, quel fichier devriez-vous modifier et quelles lignes devriez-vous ajouter?
3. Quelles précautions prendre avant de démonter un système de fichiers pour éviter la perte de données?

#### Pour aller plus loin:

* Configurez le montage automatique d'un disque réseau ou d'une partition supplémentaire via le fichier **/etc/fstab**. Testez les configurations en redémarrant votre machine et assurez-vous que tout se monte comme prévu. Documentez les étapes, les commandes utilisées, et les raisons de vos choix de configuration.

### **Exercice sur la Commande umount**

#### Objectifs:

* Apprendre à démonter de manière sécurisée les systèmes de fichiers sous Linux, une compétence essentielle pour la gestion correcte des supports de stockage.

#### Introduction:

La commande **umount** est utilisée pour démonter des systèmes de fichiers qui ont été précédemment montés avec la commande **mount**. Démonter correctement les systèmes de fichiers est crucial pour prévenir la corruption des données et assurer que toutes les données écrites sur le disque sont sécurisées avant de retirer le support de stockage.

#### Questions de Réflexion:

1. Pourquoi est-il important de démonter proprement un système de fichiers avant de déconnecter un périphérique de stockage?
2. Quels risques court-on si un périphérique est débranché sans être préalablement démonté ?
3. Comment pouvez-vous vérifier que toutes les données ont été correctement écrites sur le périphérique avant de le démonter ?

#### Exercice Pratique:

**Partie A: Exploration de umount**

1. Ouvrez un terminal sur votre système Linux.
2. Assurez-vous qu'une clé USB ou un autre périphérique de stockage est monté, comme dans l'exercice précédent.
3. Avant de démonter, utilisez la commande **sync** pour synchroniser les données sur le disque, assurant que toutes les écritures en attente sont terminées.
4. Démontez le périphérique avec **sudo umount /mnt/usb** (assurez-vous d'ajuster le chemin si nécessaire).
5. Vérifiez que le démontage a été réussi en utilisant **df -h** pour confirmer que le périphérique n'apparaît plus dans la liste des systèmes de fichiers montés.

**Partie B: Questions spécifiques**

1. Que faire si **umount** indique que le périphérique est occupé? Comment procéder pour identifier et arrêter les processus qui utilisent ce périphérique?
2. Expliquez l'importance de la commande **sync** dans le contexte du démontage d'un système de fichiers.
3. Discutez des implications de démonter un système de fichiers qui contient des sous-volumes ou des liens symboliques vers d'autres parties du système de fichiers.

#### Pour aller plus loin:

* Réalisez un script qui monte automatiquement un périphérique, copie un ensemble de fichiers vers celui-ci, puis le démonte proprement, tout en vérifiant que chaque étape s'est déroulée sans erreurs. Utilisez des commandes comme **echo $?** pour vérifier le succès des commandes dans votre script.