MEWO

Module 03 :

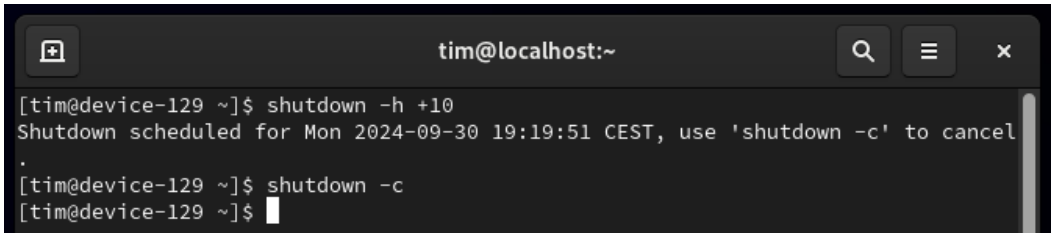
Exploiter les systèmes en fonctionnement

Timothee SICCHIA
11/10/2024

Table des matières

1. Démarrer, redémarrer et éteindre un système normalement

- **Explication** : Savoir comment démarrer, redémarrer ou éteindre un système Linux est une compétence de base pour tout administrateur. Les commandes utilisées sont simples mais essentielles pour gérer un système de manière sûre.
- ***Exemple*** :
    - Démarrer le système : C'est effectué automatiquement via le BIOS ou l'UEFI. Rien à faire de spécifique ici.
    - Pour redémarrer un système via le terminal : utilisez `sudo reboot`. Cette commande redémarre immédiatement la machine.
    - Pour éteindre le système : utilisez `sudo shutdown -h now` pour éteindre immédiatement ou `sudo shutdown -h +10` pour éteindre dans 10 minutes. Ces commandes assurent un arrêt propre du système, évitant la perte de données.



Shutdown -c permet d'annuler l'extinction prévue dans 10 minutes

2. Démarrer manuellement des systèmes sur différentes cibles

- **Explication** : Il est parfois nécessaire de démarrer un système sur une cible différente (par exemple, en mode de récupération ou mode monoutilisateur) pour résoudre des problèmes ou effectuer des tâches administratives.
- ***Exemple*** :
    - Accéder au mode de récupération : Au démarrage, maintenez la touche `Shift` ou `Esc` (selon la configuration) pour accéder au menu GRUB. Ensuite, sélectionnez le mode de récupération (`Rescue Mode`) ou monoutilisateur (`Single User Mode`). Cela permet d'effectuer des opérations de maintenance lorsque le système ne démarre pas normalement.

Caractéristique	Mode de récupération	Mode monoutilisateur
Objectif principal	Réparer des erreurs système graves ou récupérer le système	Effectuer des tâches d'administration de base ou maintenance
Services démarrés	Ensemble minimal de services, souvent sans réseau	Aucun service réseau démarré, uniquement les services de base
Accès aux utilisateurs	Multi-utilisateur limité (mais avec des restrictions)	Un seul utilisateur (root), pas de connexion utilisateur standard

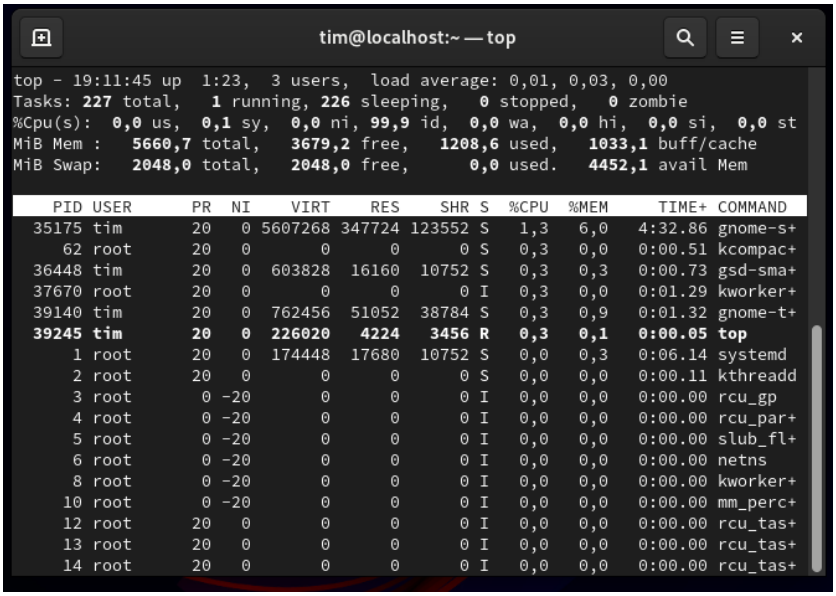
Accès réseau	Généralement sans accès réseau, mais peut être configuré	Aucun accès réseau par défaut
Cas d'utilisation	Résoudre des problèmes de démarrage, de configuration, ou de système de fichiers	Tâches administratives comme changer le mot de passe root, réparer des fichiers systèmes
Montage des systèmes de fichiers	Les systèmes de fichiers peuvent être montés en lecture seule pour éviter des dommages	Systèmes de fichiers montés, mais un seul utilisateur root a accès
Comment l'accéder	Sélectionner l'option de récupération au démarrage (GRUB)	Démarrage avec l'option `single` depuis GRUB

3. Interrompre le processus de démarrage pour accéder à un système

- **Explication** : Parfois, il est nécessaire d'interrompre le processus de démarrage pour effectuer un dépannage ou accéder à un système dont vous avez perdu le mot de passe.
- ***Exemple*** :
    - Pendant le démarrage, appuyez rapidement sur `Esc` ou `Shift` pour afficher le menu GRUB. Sélectionnez ensuite une entrée de démarrage, appuyez sur la touche `e` pour éditer les options de démarrage, et modifiez les lignes pour démarrer en mode de récupération ou changer le mot de passe root. Ceci est utile pour regagner l'accès à un système verrouillé ou défaillant.

4. Identifier et arrêter les processus intensifs en processeur et mémoire

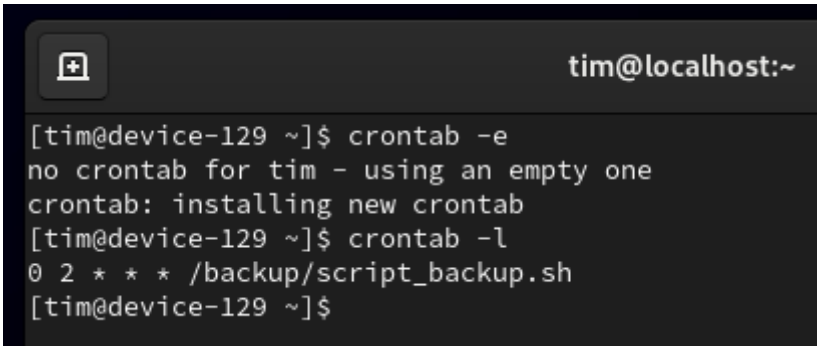
- **Explication** : Il est important de savoir identifier les processus qui consomment beaucoup de ressources et de les arrêter si nécessaire, pour optimiser les performances du système.
- ***Exemple*** :
    - Utilisez la commande `top` pour afficher les processus actifs et identifier ceux qui consomment le plus de CPU ou de mémoire. Une fois identifié, vous pouvez arrêter un processus gourmand avec `sudo kill [PID]`, où `[PID]` est l'ID du processus affiché par top. Par exemple, si un processus avec `PID 1234` consomme trop de ressources, exécutez `sudo kill 1234` pour le terminer.



La liste des processus à la suite de la commande `top`

5. Configurer la planification des tâches

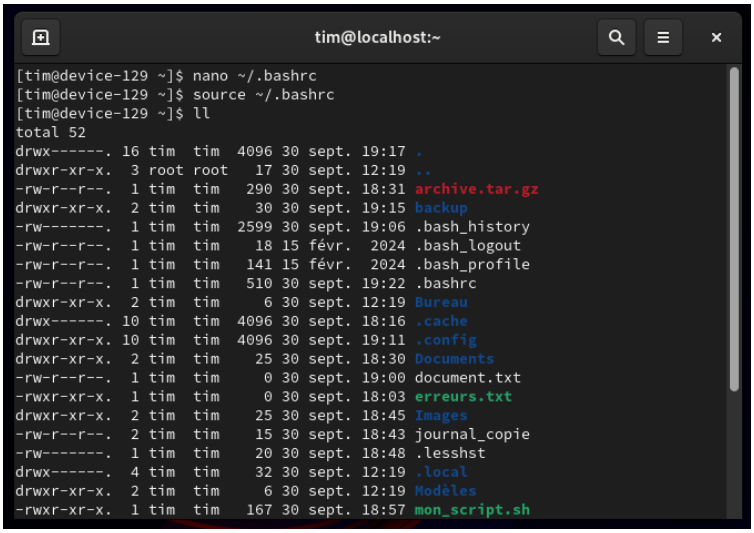
- **Explication** : La planification des tâches permet d'automatiser l'exécution de scripts ou de commandes à des intervalles réguliers ou à des moments spécifiques.
- ***Exemple*** :
    - Utilisez `crontab -e` pour éditer les tâches planifiées. Par exemple, ajoutez 0 2 * * * /usr/bin/backup.sh pour exécuter un script de sauvegarde chaque jour à 2h du matin. Cela facilite l'automatisation de tâches de maintenance sans intervention humaine.



`Crontab -l` pour afficher les tâches dans cron

6. Gérer des profils personnalisés

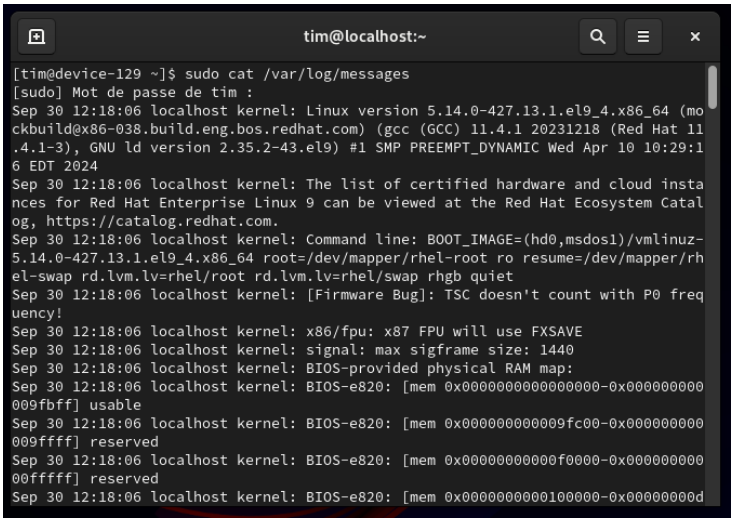
- **Explication** : Un profil personnalisé permet de configurer des paramètres spécifiques à un utilisateur ou à un environnement.
- ***Exemple*** :
    - Modifiez le fichier ~/.bashrc pour personnaliser votre profil shell. Par exemple, ajoutez alias `ll='ls -la'` pour créer un raccourci qui affiche une liste détaillée des fichiers lorsque vous tapez `ll`. Rechargez le profil avec `source ~/.bashrc` pour appliquer les changements immédiatement.



Désormais la commande `ll` exécute `ls -la`

7. Localiser et interpréter les fichiers journaux et l'historique du système

- **Explication** : Les fichiers journaux contiennent des enregistrements d'événements et d'activités du système, utiles pour le dépannage et la surveillance.
- ***Exemple*** :
    - Accédez au fichier de journal principal avec `sudo cat /var/log/ messages` ou `sudo tail -f /var/log/messages` pour surveiller les activités du système en temps réel. Cela permet d'identifier les problèmes ou de comprendre le comportement du système à des moments donnés.



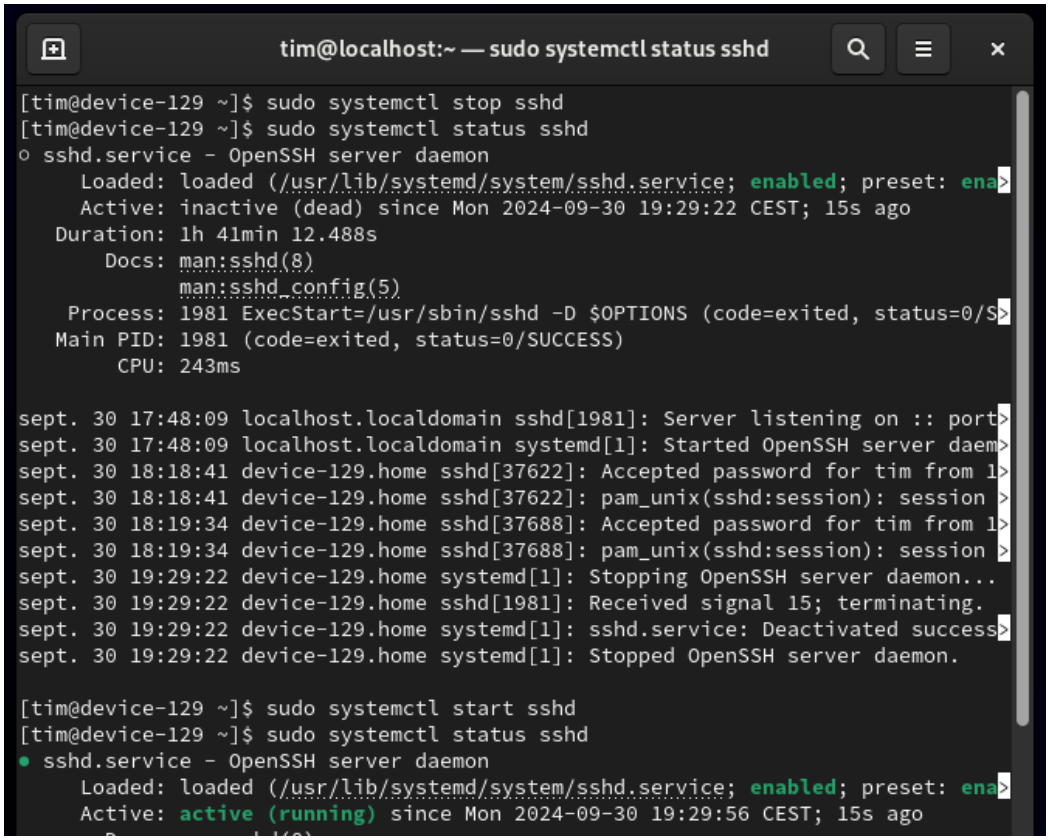
8. Conserver l'historique du système

- **Explication** : Conserver l'historique des commandes permet de suivre les actions effectuées sur un système, ce qui est crucial pour le dépannage ou la sécurité.
- ***Exemple*** :
    - L'historique des commandes exécutées est enregistré dans le fichier ~/.bash_history. Vous pouvez l'afficher en tapant `history` dans le terminal. Pour

sauvegarder l'historique manuellement, utilisez `history > historique.txt` afin de conserver un enregistrement de toutes les commandes précédentes.

9. Démarrer, arrêter et vérifier l'état des services réseau

- **Explication** : Les services réseau sont des applications ou processus qui permettent des communications sur un réseau. Il est essentiel de savoir les gérer.
- **Exemple** :
    - Pour gérer un service comme ssh, utilisez `sudo systemctl start sshd` pour le démarrer, `sudo systemctl stop sshd` pour l'arrêter, et `sudo systemctl status sshd` pour vérifier son état. Cela est essentiel pour contrôler les services disponibles sur un système.



10. Transférer des fichiers de manière sécurisée entre plusieurs systèmes

- **Explication** : Le transfert sécurisé de fichiers est important pour protéger les données lorsqu'elles sont déplacées entre systèmes.
- ***Exemple*** :
    - Utilisez `scp` (Secure Copy Protocol) faisant partie du paquet OpenSSH pour transférer des fichiers de manière sécurisée. Par exemple, `scp fichier.txt utilisateur@serveur:/chemin/destination` copie fichier.txt vers le serveur distant dans le répertoire indiqué. C'est une méthode fiable pour déplacer des fichiers entre des systèmes Linux tout en assurant la sécurité des données.