

MEWO

# Module 03 :

Exploiter les systèmes en fonctionnement

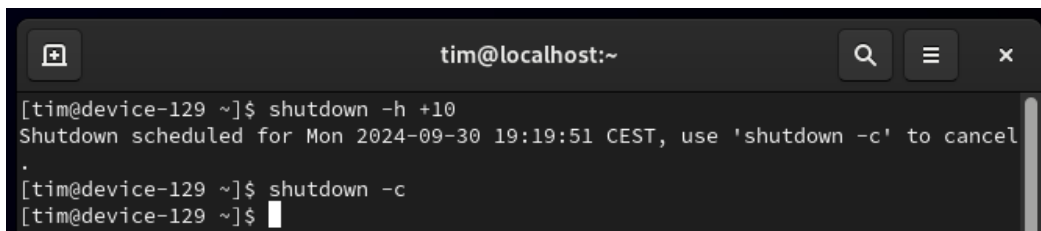
Timothee SICCHIA  
11/10/2024

## Table des matières

1. Démarrer, redémarrer et éteindre un système normalement .....	2
2. Démarrer manuellement des systèmes sur différentes cibles.....	2
3. Interrompre le processus de démarrage pour accéder à un système .....	3
4. Identifier et arrêter les processus intensifs en processeur et mémoire .....	3
5. Configurer la planification des tâches .....	4
6. Gérer des profils personnalisés .....	4
7. Localiser et interpréter les fichiers journaux et l'historique du système .....	5
8. Conserver l'historique du système.....	5
9. Démarrer, arrêter et vérifier l'état des services réseau.....	6
10. Transférer des fichiers de manière sécurisée entre plusieurs systèmes .....	6

## 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.



```

tim@localhost:~
[tim@device-129 ~]$ shutdown -h +10
Shutdown scheduled for Mon 2024-09-30 19:19:51 CEST, use 'shutdown -c' to cancel
.
[tim@device-129 ~]$ shutdown -c
[tim@device-129 ~]$
  
```

*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 monutilisateur) 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 monutilisateur (Single User Mode). Cela permet d'effectuer des opérations de maintenance lorsque le système ne démarre pas normalement.

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

<b>Accès réseau</b>	Généralement sans accès réseau, mais peut être configuré	Aucun accès réseau par défaut
<b>Cas d'utilisation</b>	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
<b>Montage des systèmes de fichiers</b>	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
<b>Comment l'accéder</b>	Sélectionner l'option de récupération au démarrage (GRUB)	Démarrage avec l'option <code>single</code> 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.

```

top - 19:11:45 up 1:23, 3 users, load average: 0,01, 0,03, 0,00
Tasks: 227 total, 1 running, 226 sleeping, 0 stopped, 0 zombie
%Cpu(s): 0,0 us, 0,1 sy, 0,0 ni, 99,9 id, 0,0 wa, 0,0 hi, 0,0 si, 0,0 st
MiB Mem : 5660,7 total, 3679,2 free, 1208,6 used, 1033,1 buff/cache
MiB Swap: 2048,0 total, 2048,0 free, 0,0 used. 4452,1 avail Mem

  PID USER      PR  NI  VIRT  RES  SHR  S  %CPU  %MEM    TIME+  COMMAND
 35175 tim        20   0 5607268 347724 123552 S   1,3   6,0   4:32.86 gnome-s+
    62 root         20   0     0     0     0 S   0,3   0,0   0:00.51 kcompac+
 36448 tim        20   0 603828 16160 10752 S   0,3   0,3   0:00.73 gsd-sma+
 37670 root        20   0     0     0     0 I   0,3   0,0   0:01.29 kworker+
 39140 tim        20   0 762456 51052 38784 S   0,3   0,9   0:01.32 gnome-t+
39245 tim        20   0 226020 4224 3456 R   0,3   0,1   0:00.05 top
    1 root        20   0 174448 17680 10752 S   0,0   0,3   0:06.14 systemd
    2 root         20   0     0     0     0 S   0,0   0,0   0:00.11 kthreadd
    3 root        -20  0     0     0     0 I   0,0   0,0   0:00.00 rcu_gp
    4 root        -20  0     0     0     0 I   0,0   0,0   0:00.00 rcu_par+
    5 root         0 -20     0     0     0 I   0,0   0,0   0:00.00 slub_fl+
    6 root        -20  0     0     0     0 I   0,0   0,0   0:00.00 netns
    8 root        -20  0     0     0     0 I   0,0   0,0   0:00.00 kworker+
   10 root         0 -20     0     0     0 I   0,0   0,0   0:00.00 mm_perc+
   12 root        20   0     0     0     0 I   0,0   0,0   0:00.00 rcu_tas+
   13 root        20   0     0     0     0 I   0,0   0,0   0:00.00 rcu_tas+
   14 root        20   0     0     0     0 I   0,0   0,0   0:00.00 rcu_tas+

```

*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.

```

[tim@device-129 ~]$ crontab -e
no crontab for tim - using an empty one
crontab: installing new crontab
[tim@device-129 ~]$ crontab -l
0 2 * * * /backup/script_backup.sh
[tim@device-129 ~]$

```

*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.

```

tim@localhost:~$ nano ~/.bashrc
[tim@device-129 ~]$ source ~/.bashrc
[tim@device-129 ~]$ ll
total 52
drwx-----. 16 tim tim 4096 30 sept. 19:17 .
drwxr-xr-x.  3 root root 17 30 sept. 12:19 ..
-rw-r--r--.  1 tim tim 290 30 sept. 18:31 archive.tar.gz
drwxr-xr-x.  2 tim tim  30 30 sept. 19:15 backup
-rw-----.  1 tim tim 2599 30 sept. 19:06 .bash_history
-rw-r--r--.  1 tim tim  18 15 févr. 2024 .bash_logout
-rw-r--r--.  1 tim tim 141 15 févr. 2024 .bash_profile
-rw-r--r--.  1 tim tim  510 30 sept. 19:22 .bashrc
drwxr-xr-x.  2 tim tim  6 30 sept. 12:19 Bureau
drwx-----. 10 tim tim 4096 30 sept. 18:16 .cache
drwxr-xr-x. 10 tim tim 4096 30 sept. 19:11 .config
drwxr-xr-x.  2 tim tim  25 30 sept. 18:30 Documents
-rw-r--r--.  1 tim tim  0 30 sept. 19:00 document.txt
-rwxr-xr-x.  1 tim tim  0 30 sept. 18:03 erreurs.txt
drwxr-xr-x.  2 tim tim  25 30 sept. 18:45 Images
-rw-r--r--.  2 tim tim  15 30 sept. 18:43 journal_copie
-rw-----.  1 tim tim  20 30 sept. 18:48 .lesshtst
drwx-----.  4 tim tim  32 30 sept. 12:19 .local
drwxr-xr-x.  2 tim tim  6 30 sept. 12:19 Modèles
-rwxr-xr-x.  1 tim tim 167 30 sept. 18:57 mon_script.sh

```

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.

```

tim@localhost:~$ sudo cat /var/log/messages
[sudo] Mot de passe de tim :
Sep 30 12:18:06 localhost kernel: Linux version 5.14.0-427.13.1.el9_4.x86_64 (mo
ckbuild@x86-038.build.eng.bos.redhat.com) (gcc (GCC) 11.4.1 20231218 (Red Hat 11
.4.1-3), GNU ld version 2.35.2-43.el9) #1 SMP PREEMPT_DYNAMIC Wed Apr 10 10:29:1
6 EDT 2024
Sep 30 12:18:06 localhost kernel: The list of certified hardware and cloud insta
nces for Red Hat Enterprise Linux 9 can be viewed at the Red Hat Ecosystem Catal
og, https://catalog.redhat.com.
Sep 30 12:18:06 localhost kernel: Command line: BOOT_IMAGE=(hd0,msdos1)/vmlinuz-
5.14.0-427.13.1.el9_4.x86_64 root=/dev/mapper/rhel-root ro resume=/dev/mapper/rh
el-swap rd.lvm.lv=rhel/root rd.lvm.lv=rhel/swap rhgb quiet
Sep 30 12:18:06 localhost kernel: [Firmware Bug]: TSC doesn't count with P0 freq
uency!
Sep 30 12:18:06 localhost kernel: x86/fpu: x87 FPU will use FXSAVE
Sep 30 12:18:06 localhost kernel: signal: max sigframe size: 1440
Sep 30 12:18:06 localhost kernel: BIOS-provided physical RAM map:
Sep 30 12:18:06 localhost kernel: BIOS-e820: [mem 0x0000000000000000-0x00000000
009fbfff] usable
Sep 30 12:18:06 localhost kernel: BIOS-e820: [mem 0x00000000000009fc00-0x00000000
009fffff] reserved
Sep 30 12:18:06 localhost kernel: BIOS-e820: [mem 0x000000000000f0000-0x00000000
00ffffff] reserved
Sep 30 12:18:06 localhost kernel: BIOS-e820: [mem 0x0000000001000000-0x00000000d

```

## 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.

```

tim@localhost:~ — sudo systemctl status sshd
[tim@device-129 ~]$ sudo systemctl stop sshd
[tim@device-129 ~]$ sudo systemctl status sshd
○ sshd.service - OpenSSH server daemon
   Loaded: loaded (/usr/lib/systemd/system/sshd.service; enabled; preset: ena>
   Active: inactive (dead) since Mon 2024-09-30 19:29:22 CEST; 15s ago
   Duration: 1h 41min 12.488s
   Docs: man:sshd(8)
         man:sshd_config(5)
   Process: 1981 ExecStart=/usr/sbin/sshd -D $OPTIONS (code=exited, status=0/S>
   Main PID: 1981 (code=exited, status=0/SUCCESS)
   CPU: 243ms

sept. 30 17:48:09 localhost.localdomain sshd[1981]: Server listening on :: port>
sept. 30 17:48:09 localhost.localdomain systemd[1]: Started OpenSSH server daem>
sept. 30 18:18:41 device-129.home sshd[37622]: Accepted password for tim from 1>
sept. 30 18:18:41 device-129.home sshd[37622]: pam_unix(sshd:session): session >
sept. 30 18:19:34 device-129.home sshd[37688]: Accepted password for tim from 1>
sept. 30 18:19:34 device-129.home sshd[37688]: pam_unix(sshd:session): session >
sept. 30 19:29:22 device-129.home systemd[1]: Stopping OpenSSH server daemon...
sept. 30 19:29:22 device-129.home sshd[1981]: Received signal 15; terminating.
sept. 30 19:29:22 device-129.home systemd[1]: sshd.service: Deactivated success>
sept. 30 19:29:22 device-129.home systemd[1]: Stopped OpenSSH server daemon.

[tim@device-129 ~]$ sudo systemctl start sshd
[tim@device-129 ~]$ sudo systemctl status sshd
● sshd.service - OpenSSH server daemon
   Loaded: loaded (/usr/lib/systemd/system/sshd.service; enabled; preset: ena>
   Active: active (running) since Mon 2024-09-30 19:29:56 CEST; 15s ago

```

## 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.