Administration Unix
4IIR
Série d'exercices

Fait par Sara Ferraa

Exercice 1 : Hiérarchie des répertoires Linux

1. Commande pour afficher les répertoires sous / : ***ls -l /***
2.

Répertoire	Description du rôle
/bin	Contient les commandes essentielles utilisables par tous les utilisateurs.
/boot	Contient les fichiers nécessaires au démarrage, comme le kernel et GRUB.
/dev	Contient les fichiers spéciaux représentant les périphériques.
/etc	Contient les fichiers de configuration du système.
home	Répertoires personnels des utilisateurs.
lib	Bibliothèques partagées nécessaires au démarrage et à l'exécution des commandes.
media	Points de montage temporaires pour des périphériques (ex. : clés USB).
mnt	Points de montage temporaires pour des systèmes de fichiers.
opt	Contient des logiciels additionnels installés manuellement.
proc	Système de fichiers virtuel contenant des informations sur les processus.
root	Répertoire personnel de l'utilisateur root.
sbin	Commandes système essentielles pour l'administration.
tmp	Contient les fichiers temporaires.
var	Contient des fichiers variables comme les logs, spool ou cache.

Exercice 2 : Démarrage et arrêt d'une machine Linux

Étape du démarrage	Fichier(s) intervenant(s)
Étape 1 : Initialisation du firmware (BIOS/UEFI)	- Firmware BIOS/UEFI - Table de partition (MBR ou GPT)
Étape 2 : Chargement du bootloader (GRUB)	- `/etc/default/grub` - `/boot/grub2/grub.cfg`

	- Commande : `grub2-mkconfig`
Étape 3 : Chargement du noyau Linux et de l'initramfs	- `/boot/vmlinuz` - `/boot/initramfs`
Étape 4 : Initialisation du système avec systemd	- `/sbin/init` (alias de systemd) - `/etc/systemd/system/default.target` - Services et unités dans `/etc/systemd/system/` et `/usr/lib/systemd/system/`
Étape 5 : Démarrage des services et de l'environnement utilisateur	- `~/.bashrc` - `/etc/profile` - Services spécifiques définis par l'utilisateur

2. Définissez ce qu'est une cible systemd et un niveau d'exécution dans un système Linux.

=>Un **niveau d'exécution** (*runlevel*) dans un système Linux représente un état prédéfini du système, définissant quels services et processus sont actifs. Les niveaux d'exécution étaient principalement utilisés dans **SysVinit**.

=>Une **cible systemd** (*target*) est l'équivalent moderne d'un niveau d'exécution, utilisé par **systemd** pour organiser le démarrage et la gestion des services. Contrairement aux niveaux d'exécution, les cibles sont plus flexibles, car elles peuvent dépendre d'autres cibles et activer plusieurs services simultanément.

Niveau d'exécution	Nom de la cible systemd	Description
0	`poweroff.target`	Arrête le système
1	`rescue.target`	Mode mono-utilisateur (maintenance)
3	`multi-user.target`	Mode multi-utilisateur en ligne de commande (sans interface graphique)
5	`graphical.target`	Mode multi-utilisateur avec interface graphique (GUI)
6	`reboot.target`	Redémarre le système

Exercice 2 : Gestion des cibles

1. **Lister les cibles de type "service" disponibles sur le système**
   ```
   systemctl list-units --type=service
   ```

2. **Afficher la cible par défaut du système**
   ```
   systemctl get-default
   ```

3. **Basculez temporairement vers la cible multi-user.target sans redémarrer le système**
   ```
   systemctl isolate multi-user.target
   ```

4. **Définir graphical.target comme cible par défaut**
   ```
   systemctl set-default graphical.target
   ```

5. **Différence entre un changement temporaire et un changement permanent de cible**

   Temporaire : Appliqué immédiatement mais non persistant après redémarrage.
   Permanent : Appliqué immédiatement et conservé après redémarrage.

Exercice 3 : Gestion des services

1. **Démarrer le service httpd (Apache)**
   ```
   systemctl start httpd
   ```

2. **Vérifier que le service httpd a bien démarré**
   ```
   systemctl status httpd
   ```

3. **Redémarrer le service sshd**
   ```
   systemctl restart sshd
   ```

4. **Cas nécessitant un redémarrage de service**

   Modification de la configuration
   Mise à jour du service
   Dysfonctionnement du service

5. **Configurer firewalld pour démarrer automatiquement**
   `systemctl enable firewalld`

6. **Vérifier si un service est activé au démarrage**
   `systemctl is-enabled <service>`

7. **Désactiver cups au démarrage**
   `systemctl disable cups`
8. **Arrêter immédiatement le service crond**
   `systemctl stop crond`

9. **Vérifier que crond est bien arrêté**
   `systemctl status crond`

10. **Masquer vsftpd pour empêcher son démarrage**
    `systemctl mask vsftpd`

11. **Vérifier qu'un service est bien masqué**
    `systemctl is-enabled vsftpd`

12. **Démasquer vsftpd**
    `systemctl unmask vsftpd`

13. **Afficher l'état complet du service nginx**
    `systemctl status nginx`

14. **Signification des états de service**
    `active (running)`: Le service est en cours d'exécution.
    `inactive (dead)`: Le service est arrêté.

Exercice 4 : Arrêt du système

1. **Arrêter immédiatement le système**
   `shutdown -h now`

2. **Programmer un arrêt dans 30 minutes**
   `shutdown -h +30`

3. **Annuler un arrêt planifié**

```
shutdown -c
```

4. **Redémarrer immédiatement le système**
```
reboot
```

5. **Différence entre `-r` et `-h`**
   `-r` : Redémarrer le système
   `-h` : Arrêter le système
6. **Mettre le système en veille prolongée**
```
systemctl hibernate
```

7. **Différences entre `suspend.target`, `hibernate.target` et `hybrid-sleep.target`**
   `suspend.target` : Mise en veille (RAM sous tension)
   `hibernate.target` : Écriture sur disque et extinction
   `hybrid-sleep.target` : Suspension + Hibernation
8. **Masquer la cible suspend.target**
```
systemctl mask suspend.target
```

Exercice 5 : Gestion des journaux avec `journalctl`

1. **Afficher tous les logs du journal système**

```
journalctl
```

2. **Afficher les logs du noyau uniquement**

```
journalctl -k
```

3. **Lister les messages du journal depuis le dernier démarrage**

```
journalctl -b
```

4. **Afficher les logs d'un service spécifique (ex: nginx)**

```
journalctl -u nginx
```

5. **Afficher les logs d'un service sur les deux derniers redémarrages**

```
journalctl -u nginx -b -2
```

6. **Afficher les logs en temps réel (suivi en direct comme `tail -f`)**

```
journalctl -f
```

7. **Filtrer les logs par date (ex: depuis le 2024-01-01)**

```
journalctl --since "2024-01-01"
```

8. **Afficher les 50 dernières lignes du journal**

```
journalctl -n 50
```

9. **Effacer entièrement les journaux du système**

```
journalctl --vacuum-size=0
```

Exercice 6 : Surveillance des processus avec top et ps

1. **Lister tous les processus en cours d'exécution**

```
ps aux
```

2. **Afficher les processus appartenant à un utilisateur spécifique (ex: root)**

```
ps -u root
```

3. **Surveiller l'activité du système en temps réel**

```
top
```

4. **Classer les processus par utilisation mémoire dans top**
   Appuyer sur **Shift + M** dans top
5. **Trouver le PID d'un processus spécifique (ex: apache2)**

```
pgrep apache2
```

6. **Afficher les processus hiérarchiquement (arborescence)**

```
pstree
```

7. **Tuer un processus par son PID (ex: 1234)**

```
kill 1234
```

8. **Forcer la fermeture d'un processus (ex: firefox)**

```
killall -9 firefox
```

Exercice 7 : Gestion des utilisateurs et permissions

1. **Créer un nouvel utilisateur (ex: alice)**

```
useradd alice
```

2. **Attribuer un mot de passe à l'utilisateur alice**

```
passwd alice
```

3. **Ajouter un utilisateur à un groupe (ex: alice au groupe sudo)**

```
usermod -aG sudo alice
```

4. **Lister les groupes auxquels appartient un utilisateur**

```
groups alice
```

5. **Modifier le shell par défaut d'un utilisateur (ex: bash)**

```
chsh -s /bin/bash alice
```

6. **Changer l’utilisateur courant vers alice**

```
su - alice
```

7. **Supprimer un utilisateur et son répertoire personnel**

```
userdel -r alice
```

8. **Vérifier les permissions d’un fichier (ex: `/etc/passwd`)**

```
ls -l /etc/passwd
```

9. **Changer les permissions d’un fichier pour que seul le propriétaire puisse lire et écrire**

```
chmod 600 fichier.txt
```

10. **Changer le propriétaire d’un fichier**

```
chown alice fichier.txt
```

Exercice 8 : Gestion du réseau avec `ip` et `netstat`

1. **Afficher la configuration réseau (adresse IP, interfaces actives)**

```
ip a
```

2. **Afficher la table de routage**

```
ip route
```

3. **Afficher les connexions réseau actives**

```
netstat -tunapl
```

4. **Tester la connectivité avec une adresse IP ou un site web**

```
ping google.com
```

5. **Afficher les ports ouverts sur la machine**

```
ss -tulnp
```

6. **Vérifier la connectivité d'un port spécifique (ex: 22)**

```
nc -zv 192.168.1.1 22
```

7. **Afficher les statistiques réseau en direct**

```
iftop
```

Exercice 9 : Gestion des disques et partitions

1. **Lister les disques et partitions disponibles**

```
lsblk
```

2. **Afficher l'espace disque utilisé et disponible**

```
df -h
```

3. **Afficher l'espace occupé par un dossier spécifique (ex: `/var/log`)**

```
du -sh /var/log
```

4. **Vérifier l'état des partitions et le type de fichiers système**

```
fdisk -l
```

5. **Monter une partition (ex: `/dev/sdb1` sur `/mnt`)**

```
mount /dev/sdb1 /mnt
```

6. **Démonter une partition**

```
umount /mnt
```

7. **Vérifier les erreurs d'un disque**

```
fsck /dev/sdb1
```

Exercice 10 : Sécurité et pare-feu avec `firewalld` et `iptables`

1. **Vérifier si le pare-feu est actif**

```
systemctl status firewalld
```

2. **Activer le pare-feu**

```
systemctl start firewalld
```

3. **Autoriser le trafic HTTP (port 80)**

```
firewall-cmd --add-service=http --permanent
```

4. **Recharger la configuration du pare-feu**

```
firewall-cmd --reload
```

5. **Lister les règles de pare-feu actives**

```
firewall-cmd --list-all
```

6. **Bloquer une adresse IP spécifique (ex: 192.168.1.100)**

```
iptables -A INPUT -s 192.168.1.100 -j DROP
```

7. **Vérifier les règles `iptables` actives**

```
iptables -L -v
```

8. **Sauvegarder la configuration actuelle du pare-feu**

```
iptables-save > /etc/iptables.rules
```