

Administration des systèmes et réseaux Linux

Niveau : GL2 (Génie Logiciel - 2ème année)

Volume horaire : 30 heures (2 crédits) - CM, TD, TP, TPE

Enseignant : ING. GUIYوبا JONATHAN

Année académique : 2025-2026

Objectifs du cours

À l'issue de ce cours, les étudiants seront capables de :

- Comprendre l'architecture et le fonctionnement du système Linux
- Maîtriser les commandes de base et d'administration système
- Gérer les utilisateurs et les permissions
- Administrer efficacement un système Linux
- Utiliser les outils de communication et de développement sous Linux

Table des matières

1. Le système Linux
 - 1.1 Historique
 - 1.2 Principales caractéristiques
 - 1.3 Architecture du système
 - 1.4 Arborescence
 - 1.5 Utilisateurs et groupes
 - 1.6 Connexion/déconnexion
 - 1.7 Protection des données
 - 1.8 Syntaxe
 - 1.9 Principales commandes
 - 1.10 Éditeur de texte
 - 1.11 Outils de communication
2. Commandes d'administration système
3. Travaux pratiques

1. Le système Linux

1.1 Historique

Linux est né en 1991 grâce à Linus Torvalds, un étudiant finlandais de 21 ans à l'université d'Helsinki. Voici les étapes clés de son développement :

1.1.1 Les origines (1991)

- **Août 1991** : Linus Torvalds annonce son projet sur le newsgroup comp.os.minix
- **Septembre 1991** : Première version (0.01) avec 10 239 lignes de code
- **Octobre 1991** : Version 0.02, première version "officielle"

1.1.2 Développement communautaire (1992-1994)

- **1992** : Adoption de la licence GPL (GNU General Public License)
- **1993** : Plus de 100 développeurs contribuent au projet
- **1994** : Version 1.0, premier noyau stable

À savoir : Le nom "Linux" vient de la combinaison de "Linus" et "Unix". Initialement, Linus voulait l'appeler "Freax" (Free + Freak + Unix).

1.1.3 Évolution moderne (1995-aujourd'hui)

- **1995-1999** : Émergence des premières distributions (Red Hat, SUSE, Debian)
- **2000s** : Adoption massive par les entreprises et les serveurs
- **2010s** : Domination du marché mobile avec Android
- **Aujourd'hui** : Présent partout : serveurs, smartphones, IoT, supercalculateurs

1.2 Principales caractéristiques

1.2.1 Caractéristiques techniques

Caractéristique	Description	Avantages
Multitâche	Exécution simultanée de plusieurs processus	Efficacité, gestion optimale des ressources
Multi-utilisateur	Plusieurs utilisateurs peuvent utiliser le système simultanément	Partage des ressources, sécurité
Portable	Fonctionne sur différentes architectures (x86, ARM, RISC-V...)	Flexibilité, large adoption
Open Source	Code source libre et modifiable	Transparence, sécurité, personnalisation
Stable	Fonctionnement continu sans redémarrage fréquent	Fiabilité pour les serveurs critiques

1.2.2 Avantages de Linux

Pourquoi choisir Linux ?

- **Gratuit** : Aucun coût de licence
- **Sécurisé** : Moins vulnérable aux virus et malwares
- **Performant** : Utilisation optimale des ressources système
- **Personnalisable** : Interface et comportement entièrement configurables
- **Communauté active** : Support et développement continus

1.2.3 Les distributions Linux populaires

Distribution	Usage principal	Caractéristiques
Ubuntu	Desktop, débutants	Facile à utiliser, large communauté
CentOS/RHEL	Serveurs d'entreprise	Stabilité, support commercial
Debian	Serveurs, développement	Très stable, base d'Ubuntu
Arch Linux	Utilisateurs avancés	Personnalisation maximale
Fedora	Développeurs, technologies récentes	Innovations, technologies de pointe

1.3 Architecture du système

1.3.1 Architecture en couches

Le système Linux est organisé en plusieurs couches, chacune ayant un rôle spécifique :



1.3.2 Le noyau Linux (Kernel)

Le noyau est le cœur du système. Il gère :

- **Gestion des processus** : Création, ordonnancement, communication
- **Gestion mémoire** : Allocation, pagination, mémoire virtuelle
- **Système de fichiers** : Accès aux fichiers et répertoires
- **Pilotes de périphériques** : Communication avec le matériel
- **Gestion réseau** : Protocoles TCP/IP, sockets

Important : Le noyau s'exécute en mode privilégié (mode noyau), tandis que les applications utilisateur s'exécutent en mode utilisateur avec des privilèges limités.

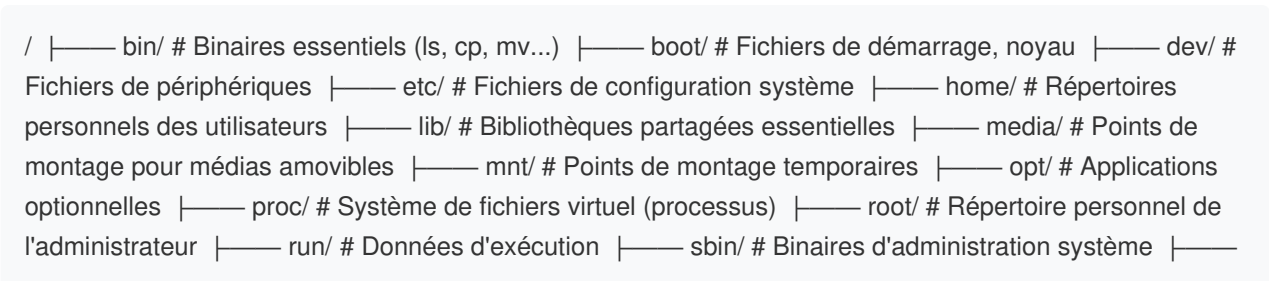
1.3.3 Les espaces utilisateur et noyau

Espace	Privilèges	Composants	Accès matériel
Noyau	Complets	Kernel, pilotes, services système	Direct
Utilisateur	Limités	Applications, bibliothèques, shell	Via appels système

1.4 Arborescence du système de fichiers

1.4.1 Structure hiérarchique

Linux utilise une structure arborescente unique, commençant par la racine (/) :



srv/ # Données des services |—— sys/ # Interface avec le noyau |—— tmp/ # Fichiers temporaires
|—— usr/ # Hiérarchie secondaire (programmes utilisateur) |—— var/ # Données variables (logs, cache...)

1.4.2 Répertoires importants détaillés

/bin - Binaires essentiels

Contient les commandes indispensables au fonctionnement du système :

ls /bin # Affiche : bash, cat, cp, date, echo, grep, ls, mv, ps, rm, sh...

/etc - Configuration système

Fichiers de configuration de tout le système :

- **/etc/passwd** : Informations sur les utilisateurs
- **/etc/group** : Définition des groupes
- **/etc/hosts** : Résolution de noms locale
- **/etc/fstab** : Systèmes de fichiers à monter
- **/etc/ssh** : Configuration SSH

/home - Répertoires utilisateurs

Chaque utilisateur a son répertoire personnel dans /home :

/home/ |—— alice/ # Répertoire de l'utilisatrice Alice |—— bob/ # Répertoire de l'utilisateur Bob |—— charlie/ # Répertoire de l'utilisateur Charlie

/var - Données variables

Données qui changent pendant le fonctionnement du système :

- **/var/log/** : Fichiers de journalisation (logs)
- **/var/cache/** : Cache des applications
- **/var/tmp/** : Fichiers temporaires persistants
- **/var/www/** : Contenu des serveurs web

Exercice 1 : Exploration de l'arborescence

Objectif : Se familiariser avec la structure des répertoires

Consignes :

1. Listez le contenu de la racine (/) et identifiez 5 répertoires
2. Explorez le contenu de /etc et trouvez le fichier de configuration des utilisateurs
3. Vérifiez la taille du répertoire /var/log

1.5 Utilisateurs et groupes d'utilisateurs

1.5.1 Concept d'utilisateur sous Linux

Linux est un système multi-utilisateur où chaque utilisateur est identifié par :

- **UID (User ID)** : Identifiant numérique unique
- **Nom d'utilisateur** : Nom lisible associé à l'UID
- **Répertoire personnel** : Espace de travail privé
- **Shell par défaut** : Interface de commande

1.5.2 Types d'utilisateurs

Type	UID	Description	Exemple
root	0	Superutilisateur, tous les privilèges	Administrateur système
Système	1-999	Comptes pour les services système	daemon, www-data, mysql
Utilisateurs normaux	1000+	Comptes d'utilisateurs réels	alice, bob, charlie

1.5.3 Gestion des utilisateurs

Fichiers de configuration

/etc/passwd : Informations de base sur les utilisateurs

```
root:x:0:0:root:/root:/bin/bash alice:x:1001:1001:Alice Martin,,,:/home/alice:/bin/bash bob:x:1002:1002:Bob Dupont,,,:/home/bob:/bin/bash
Format : nom:mot_de_passe:UID:GID:commentaire:répertoire:shell
```

/etc/shadow : Mots de passe chiffrés (accessible uniquement par root)

```
root:$6$randomsalt$hashedpassword:18000:0:99999:7:::
alice:$6$anothersalt$anotherhashedpassword:18000:0:99999:7:::
Format :
nom:mot_de_passe_chiffré:dernière_modif:min:max:avertissement:inactivité:expiration
```

Commandes de gestion des utilisateurs

```
# Créer un utilisateur sudo useradd -m -s /bin/bash nouveauuser # Définir un mot de passe sudo passwd nouveauuser # Modifier les informations utilisateur sudo usermod -c "Nom Complet" -s /bin/zsh utilisateur # Supprimer un utilisateur sudo userdel -r utilisateur # Changer d'utilisateur su - autreuser
```

1.5.4 Groupes d'utilisateurs

Concept de groupe

Les groupes permettent de :

- Organiser les utilisateurs par fonction ou projet
- Simplifier la gestion des permissions
- Partager des ressources entre utilisateurs

/etc/group : Définition des groupes

```
root:x:0: users:x:100:alice,bob developers:x:1001:alice,charlie admins:x:1002:alice Format :  
nom_groupe:mot_de_passe:GID:liste_des_membres
```

Commandes de gestion des groupes

```
# Créer un groupe sudo groupadd developers # Ajouter un utilisateur à un groupe sudo usermod -a -G  
developers alice # Afficher les groupes d'un utilisateur groups alice # Changer le groupe principal d'un  
utilisateur sudo usermod -g newgroup alice # Supprimer un groupe sudo groupdel oldgroup
```

Conseil pratique : Utilisez la commande `id` pour afficher toutes les informations d'identité d'un utilisateur (UID, GID, groupes).

1.6 Connexion/Déconnexion

1.6.1 Processus de connexion

Connexion locale

1. **Démarrage** : Le système démarre et lance getty sur les terminaux
2. **Invite de connexion** : L'utilisateur saisit son nom d'utilisateur
3. **Authentification** : Saisie et vérification du mot de passe
4. **Lancement du shell** : Ouverture de la session utilisateur

```
Ubuntu 20.04.3 LTS server01 tty1 server01 login: alice Password: Welcome to Ubuntu 20.04.3 LTS  
(GNU/Linux 5.4.0-88-generic x86_64) Last login: Mon Oct 25 09:15:23 2021 from 192.168.1.100  
alice@server01:~$
```

Connexion à distance (SSH)

```
# Connexion SSH simple ssh alice@192.168.1.100 # Connexion SSH avec port spécifique ssh -p 2222  
alice@192.168.1.100 # Connexion SSH avec clé privée ssh -i ~/.ssh/ma_cle alice@192.168.1.100 #  
Connexion SSH avec redirection de port ssh -L 8080:localhost:80 alice@192.168.1.100
```

1.6.2 Gestion de session

Variables d'environnement importantes

```
# Afficher l'utilisateur actuel whoami # Afficher les informations de session who w # Variables  
d'environnement de session echo $USER # Nom d'utilisateur echo $HOME # Répertoire personnel echo  
$PATH # Chemins de recherche des commandes echo $SHELL # Shell par défaut echo $PWD #  
Répertoire courant
```

Fichiers de configuration de session

Fichier	Portée	Moment d'exécution	Usage
<code>/etc/profile</code>	Tous les utilisateurs	Connexion	Configuration système globale
<code>~/.profile</code>	Utilisateur	Connexion	Configuration personnelle
<code>~/.bashrc</code>	Utilisateur (bash)	Ouverture shell	Alias, fonctions, prompt
<code>~/.bash_logout</code>	Utilisateur (bash)	Déconnexion	Nettoyage, sauvegarde

1.6.3 Déconnexion et fermeture de session

```
# Déconnexion simple exit # Déconnexion avec code de sortie exit 0 # Déconnexion forcée (CTRL+D) #
Raccourci clavier pour envoyer EOF au shell # Fermer toutes les sessions d'un utilisateur sudo pkill -u alice
# Redémarrer le système sudo reboot # Arrêter le système sudo shutdown -h now
```

Attention : Toujours se déconnecter proprement pour éviter la perte de données et libérer les ressources système.

1.7 Protection des données

1.7.1 Système de permissions Unix

Linux utilise un système de permissions basé sur trois niveaux :

Niveau	Description	Symbole	Position
Propriétaire (User)	Utilisateur qui possède le fichier	u	Positions 2-4
Groupe (Group)	Groupe propriétaire du fichier	g	Positions 5-7
Autres (Others)	Tous les autres utilisateurs	o	Positions 8-10

1.7.2 Types de permissions

Permission	Symbole	Valeur octale	Fichier	Répertoire
Lecture	r	4	Lire le contenu	Lister le contenu
Écriture	w	2	Modifier le contenu	Créer/supprimer des fichiers
Exécution	x	1	Exécuter le fichier	Accéder au répertoire

1.7.3 Lecture des permissions

```
$ ls -l /home/alice/documents/ -rw-r--r-- 1 alice users 1234 Oct 25 10:30 rapport.txt drwxr-xr-x 2 alice users 4096 Oct 25 09:15 projets/ -rwx----- 1 alice alice 256 Oct 24 16:45 script.sh
```

Décomposition : - Premier caractère : type (- = fichier, d = répertoire, l = lien...) - Caractères 2-4 : permissions du propriétaire (rw-) - Caractères 5-7 : permissions du groupe (r--) - Caractères 8-10 : permissions des autres (r--)

1.7.4 Modification des permissions

Méthode symbolique avec chmod

```
# Ajouter la permission d'exécution au propriétaire chmod u+x script.sh # Retirer la permission d'écriture au groupe chmod g-w fichier.txt # Donner toutes les permissions au propriétaire, lecture au reste chmod u=rwx,go=r fichier.txt # Ajouter la permission de lecture à tous chmod a+r fichier.txt
```

Méthode octale avec chmod

```
# rwx rwx rwx = 777 (toutes les permissions) chmod 777 fichier.txt # rw- r-- r-- = 644 (lecture/écriture propriétaire, lecture autres) chmod 644 document.txt # rwx r-x r-x = 755 (exécution propriétaire, lecture/exécution autres) chmod 755 script.sh # rw- --- --- = 600 (lecture/écriture propriétaire uniquement) chmod 600 fichier_privé.txt
```

Permissions courantes :

- **644** : Fichiers texte standard

- **755** : Scripts exécutables, répertoires
- **600** : Fichiers privés (mots de passe, clés)
- **700** : Répertoires privés

1.7.5 Changement de propriétaire

```
# Changer le propriétaire sudo chown alice fichier.txt # Changer le propriétaire et le groupe sudo chown
alice:developers fichier.txt # Changer récursivement (répertoires) sudo chown -R alice:users
/home/alice/projets/ # Changer seulement le groupe sudo chgrp developers fichier.txt
```

1.7.6 Permissions spéciales

Permission	Symbole	Octal	Description
SetUID	s (user)	4000	Exécute avec les privilèges du propriétaire
SetGID	s (group)	2000	Exécute avec les privilèges du groupe
Sticky Bit	t	1000	Seul le propriétaire peut supprimer (répertoires)

```
# Exemple : /tmp avec sticky bit ls -ld /tmp drwxrwxrwt 12 root root 4096 Oct 25 10:30 /tmp # Appliquer le
sticky bit sudo chmod +t /shared/directory sudo chmod 1755 /shared/directory
```

1.8 Syntaxe des commandes Linux

1.8.1 Structure générale d'une commande

```
commande [options] [arguments] Exemples : ls # Commande simple ls -l # Commande avec option ls -l
/home # Commande avec option et argument ls -la /home/alice # Commande avec options multiples et
argument
```

1.8.2 Types d'options

Options courtes (un caractère)

```
ls -l # Option longue (-l) ls -a # Option cachée (-a) ls -la # Options combinées ls -l -a # Options séparées
(équivalent)
```

Options longues (mots complets)

```
ls --long # Équivalent à -l ls --all # Équivalent à -a ls --human-readable # Option spécifique aux options
longues grep --color=auto # Option avec valeur
```

1.8.3 Caractères spéciaux du shell

Caractère	Nom	Fonction	Exemple
*	Astérisque	Joker (0 ou plusieurs caractères)	ls *.txt
?	Point d'interrogation	Joker (un caractère)	ls fichier?.txt
[]	Crochets	Classe de caractères	ls [abc]*.txt
	Pipe	Redirection de sortie	ls grep txt
>	Redirection	Sortie vers fichier	ls > liste.txt
&	Esperluette	Exécution en arrière-plan	ping google.com &

1.8.4 Redirection et pipes

Redirection de sortie

```
# Rediriger la sortie vers un fichier (écrasement) echo "Hello World" > fichier.txt # Rediriger la sortie vers un
fichier (ajout) echo "Nouvelle ligne" >> fichier.txt # Rediriger les erreurs ls /repertoire/inexistant 2>
erreurs.log # Rediriger sortie standard ET erreurs commande > output.txt 2>&1
```

Redirection d'entrée

```
# Lire depuis un fichier sort < liste.txt # Utiliser un document en ligne (heredoc) cat << EOF >
nouveau_fichier.txt Ligne 1 Ligne 2 EOF
```

Pipes (tubes)

```
# Enchaîner des commandes ls -la | grep "^d" # Lister seulement les répertoires ps aux | grep apache | wc -
l # Compter les processus Apache cat /var/log/access.log | tail -100 | grep ERROR # Dernières erreurs
```

1.8.5 Historique des commandes

```
# Afficher l'historique history # Exécuter la dernière commande !! # Exécuter la commande numéro 15 de
l'historique !15 # Rechercher dans l'historique (CTRL+R) # Tapez CTRL+R puis commencez à taper la
commande # Nettoyer l'historique history -c
```

1.9 Principales commandes utilisées

1.9.1 Navigation dans le système de fichiers

```
# Afficher le répertoire courant pwd # Changer de répertoire cd /home/alice cd .. # Répertoire parent cd ~ #
Répertoire personnel cd - # Répertoire précédent # Lister le contenu ls # Liste simple ls -l # Liste détaillée ls
-la # Inclure les fichiers cachés ls -lh # Tailles lisibles par l'homme ls -lt # Trier par date de modification ls -ls
# Trier par taille
```

1.9.2 Manipulation de fichiers et répertoires

```
# Créer des répertoires mkdir nouveau_dossier mkdir -p chemin/vers/nouveau/dossier # Créer les parents si nécessaire # Copier des fichiers cp fichier.txt copie.txt cp -r dossier/ nouveau_dossier/ # Copie récursive # Déplacer/renommer mv ancien_nom nouveau_nom mv fichier.txt /autre/repertoire/ # Supprimer rm fichier.txt rm -r dossier/ # Suppression récursive rm -rf dossier/ # Force la suppression
```

Attention : La commande `rm -rf` supprime définitivement les fichiers. Il n'y a pas de corbeille sous Linux !

1.9.3 Consultation et recherche

```
# Afficher le contenu d'un fichier cat fichier.txt # Affichage complet less fichier.txt # Navigation page par page head fichier.txt # 10 premières lignes tail fichier.txt # 10 dernières lignes tail -f /var/log/syslog # Suivi en temps réel # Rechercher des fichiers find /home -name "*.txt" # Par nom find /var -size +100M # Par taille find /etc -mtime -7 # Modifiés dans les 7 derniers jours # Rechercher dans le contenu grep "motif" fichier.txt grep -r "erreur" /var/log/ # Recherche récursive grep -i "error" fichier.txt # Insensible à la casse
```

1.9.4 Informations système

```
# Informations sur le système uname -a # Informations kernel lsb_release -a # Version de la distribution whoami # Utilisateur courant id # UID, GID et groupes groups # Groupes de l'utilisateur # Utilisation du disque df -h # Espace disque du -sh /home/alice # Taille d'un répertoire # Processus ps aux # Liste tous les processus ps aux | grep apache # Processus spécifiques top # Surveillance temps réel htop # Version améliorée de top
```

1.9.5 Gestion des processus

```
# Lancer un processus en arrière-plan commande & # Arrêter un processus kill PID # Arrêt normal kill -9 PID # Arrêt forcé killall nom_processus # Arrêter par nom # Gestion des jobs jobs # Lister les jobs fg %1 # Ramener le job 1 au premier plan bg %1 # Envoyer le job 1 en arrière-plan nohup commande & # Processus persistant après déconnexion
```

Exercice 2 : Manipulation de fichiers

Objectif : Maîtriser les commandes de base

Consignes :

1. Créez un répertoire "tp_linux" dans votre home
2. Créez un fichier "info.txt" contenant votre nom et la date
3. Copiez ce fichier sous le nom "sauvegarde.txt"
4. Listez tous les fichiers avec permissions détaillées
5. Trouvez tous les fichiers .txt dans votre répertoire personnel

1.10 Éditeur de texte

1.10.1 Éditeurs disponibles sous Linux

Éditeur	Type	Complexité	Usage typique
nano	Simple	Facile	Édition rapide, débutants
vim/vi	Modal	Complexe	Développement, administration
emacs	Complet	Complexe	Développement avancé
gedit	Graphique	Simple	Environnement de bureau

1.10.2 Nano - L'éditeur simple

Utilisation de base

```
# Ouvrir un fichier avec nano nano fichier.txt # Ouvrir un nouveau fichier nano nouveau_fichier.txt # Ouvrir avec numérotation des lignes nano -l fichier.txt
```

Raccourcis clavier nano

Raccourci	Action	Description
Ctrl+X	Quitter	Fermer nano (demande sauvegarde)
Ctrl+O	Sauvegarder	Écrire le fichier
Ctrl+W	Rechercher	Chercher du texte
Ctrl+K	Couper ligne	Couper la ligne courante
Ctrl+U	Coller	Coller le texte coupé
Ctrl+G	Aide	Afficher l'aide

1.10.3 Vi/Vim - L'éditeur modal

Les modes de vim

- **Mode commande** : Navigation et commandes (mode par défaut)
- **Mode insertion** : Saisie de texte
- **Mode visuel** : Sélection de texte
- **Mode commande ex** : Commandes avancées (: et /)

Commandes de base vim

```
# Ouvrir un fichier vim fichier.txt # Modes i # Passer en mode insertion (avant le curseur) a # Mode insertion (après le curseur) o # Nouvelle ligne et mode insertion Esc # Retour au mode commande # Navigation (mode commande) h, j, k, l # Gauche, bas, haut, droite w # Mot suivant b # Mot précédent 0 # Début de ligne $ # Fin de ligne gg # Début du fichier G # Fin du fichier # Sauvegarde et sortie :w # Sauvegarder :q # Quitter :wq # Sauvegarder et quitter :q! # Quitter sans sauvegarder
```

Commandes d'édition vim

```
# Suppression x # Supprimer caractère sous le curseur dd # Supprimer ligne courante dw # Supprimer mot d$ # Supprimer jusqu'à la fin de ligne # Copier/coller yy # Copier ligne courante yw # Copier mot p # Coller après le curseur P # Coller avant le curseur # Recherche et remplacement /motif # Rechercher "motif" n # Occurrence suivante N # Occurrence précédente :%s/ancien/nouveau/g # Remplacer partout
```

Conseil : Pour les débutants, commencez par nano. Vim est très puissant mais demande un apprentissage plus long.

1.11 Outils de communication

1.11.1 Communication locale

Entre utilisateurs du système

```
# Voir les utilisateurs connectés who w users # Envoyer un message à un utilisateur write alice # Tapez votre message, puis Ctrl+D pour envoyer # Envoyer un message à tous les utilisateurs connectés wall "Le système va redémarrer dans 10 minutes" # Envoyer un message et le logger logger "Message important dans les logs système"
```

Communication par fichiers

```
# Laisser un message dans un fichier partagé echo "Réunion demain à 14h" >> /shared/messages.txt # Surveiller un fichier de messages tail -f /shared/messages.txt
```

1.11.2 Communication réseau

SSH - Secure Shell

```
# Connexion SSH basique ssh user@hostname # SSH avec port spécifique ssh -p 2222 user@hostname # Copie de fichiers avec SCP scp fichier.txt user@hostname:/remote/path/ scp user@hostname:/remote/file.txt ./ # Synchronisation avec rsync rsync -av /local/dir/ user@hostname:/remote/dir/
```

Tests de connectivité

```
# Test de connectivité réseau ping google.com ping -c 4 192.168.1.1 # 4 pings seulement # Test de port telnet hostname 80 nc -zv hostname 22 # Test port SSH # Informations réseau ifconfig # Configuration réseau (obsolète) ip addr show # Configuration réseau (moderne) netstat -tuln # Ports en écoute ss -tuln # Version moderne de netstat
```

1.11.3 Outils de transfert de fichiers

```
# FTP sécurisé avec SFTP sftp user@hostname # Commands SFTP: # get remote_file local_file # put
local_file remote_file # ls, cd, pwd, mkdir, rm # Wget - Téléchargement web wget
https://example.com/file.zip wget -r https://example.com/ # Téléchargement récursif # Curl - Client web
polyvalent curl -O https://example.com/file.zip curl -X POST -d "data=value" https://api.example.com/
```

2. Les commandes d'administration système

2.1 Gestion des utilisateurs et groupes (avancée)

2.1.1 Commandes d'administration des utilisateurs

```
# Créer un utilisateur avec options spécifiques sudo useradd -m -s /bin/bash -G sudo,developers -c "Alice Martin" alice # Options utiles de useradd : # -m : créer le répertoire personnel # -s : définir le shell par défaut # -G : groupes supplémentaires # -c : commentaire/nom complet # -e : date d'expiration # -f : jours d'inactivité avant blocage # Modifier un utilisateur existant sudo usermod -aG admin alice # Ajouter à un groupe sudo usermod -s /bin/zsh alice # Changer le shell sudo usermod -l alice_new alice # Changer le nom d'utilisateur # Verrouiller/déverrouiller un compte sudo usermod -L alice # Verrouiller sudo usermod -U alice # Déverrouiller # Forcer le changement de mot de passe à la prochaine connexion sudo passwd -e alice
```

2.1.2 Gestion avancée des mots de passe

```
# Politique de mots de passe sudo chage alice # Modifier la politique interactivement sudo chage -M 90 alice # Mot de passe expire dans 90 jours sudo chage -W 7 alice # Avertissement 7 jours avant expiration sudo chage -I 30 alice # Compte inactif après 30 jours # Voir les informations d'expiration sudo chage -l alice # Générer un mot de passe aléatoire pwgen 12 1 # Mot de passe de 12 caractères openssl rand -base64 12 # Mot de passe cryptographiquement sûr
```

2.1.3 Surveillance des connexions

```
# Voir les utilisateurs connectés who # Utilisateurs actuellement connectés w # Informations détaillées last # Historique des connexions last alice # Connexions d'Alice uniquement lastb # Tentatives de connexion échouées # Logs d'authentification sudo tail -f /var/log/auth.log # Surveillance en temps réel sudo grep "Failed password" /var/log/auth.log # Tentatives échouées
```

2.2 Gestion des processus

2.2.1 Surveillance des processus

```
# Lister tous les processus ps aux # Format BSD ps -ef # Format System V pstree # Arbre des processus pstree -p # Avec les PID # Surveillance temps réel top # Surveillance basique htop # Interface améliorée atop # Surveillance avancée avec historique # Informations détaillées sur un processus ps -fp PID # Informations complètes ls -la /proc/PID/ # Répertoire proc du processus cat /proc/PID/status # Statut détaillé
```

2.2.2 Gestion des signaux

Signal	Numéro	Action par défaut	Usage
SIGTERM	15	Terminaison propre	Arrêt normal d'un processus
SIGKILL	9	Terminaison forcée	Arrêt immédiat (non capturable)
SIGHUP	1	Rechargement	Recharger la configuration
SIGSTOP	19	Pause	Suspendre le processus
SIGCONT	18	Continuation	Reprendre le processus suspendu

```
# Envoyer des signaux kill PID # SIGTERM par défaut kill -9 PID # SIGKILL (forcé) kill -HUP PID # SIGHUP
(rechargement) kill -STOP PID # Suspendre kill -CONT PID # Reprendre # Tuer par nom de processus
killall firefox # Tous les processus Firefox pkill -f "python script.py" # Par ligne de commande complète #
Lister les signaux disponibles kill -l # Liste de tous les signaux
```

2.2.3 Gestion des priorités

```
# Valeurs de nice : -20 (priorité maximale) à +19 (priorité minimale) # Seul root peut utiliser des valeurs
négatives # Lancer avec priorité spécifique nice -n 10 commande_gourmande # Priorité basse nice -n -5
commande_critique # Priorité haute (root uniquement) # Modifier la priorité d'un processus existant renice 5
PID # Réduire la priorité renice -10 PID # Augmenter la priorité (root) renice +10 -u alice # Réduire priorité
de tous les processus d'Alice # Voir les priorités ps -eo pid,ppid,ni,comm # Afficher les valeurs nice top #
Colonne NI pour nice, PR pour priorité
```

2.3 Gestion des services système (systemd)

2.3.1 Introduction à systemd

Systemd est le gestionnaire de services moderne de Linux qui remplace les anciens systèmes SysV init. Il gère :

- Le démarrage et l'arrêt des services
- La gestion des dépendances
- Les journaux système
- Les tâches planifiées (timers)
- Le montage automatique des volumes

2.3.2 Commandes systemctl

```
# Gestion des services sudo systemctl start apache2 # Démarrer un service sudo systemctl stop apache2 #
Arrêter un service sudo systemctl restart apache2 # Redémarrer un service sudo systemctl reload apache2
# Recharger la configuration sudo systemctl status apache2 # Voir le statut d'un service # Gestion de
l'activation sudo systemctl enable apache2 # Activer au démarrage sudo systemctl disable apache2 #
Désactiver au démarrage sudo systemctl is-enabled apache2 # Vérifier si activé # Lister les services
systemctl list-units --type=service # Tous les services systemctl list-units --failed # Services en échec
systemctl list-unit-files --type=service # Fichiers de configuration
```

2.3.3 Analyse des logs avec journalctl

```
# Voir tous les logs journalctl # Logs d'un service spécifique journalctl -u apache2 journalctl -u apache2 --
since today journalctl -u apache2 --since "2024-01-01" --until "2024-01-02" # Suivi en temps réel journalctl -f
# Tous les logs journalctl -u apache2 -f # Logs d'un service # Filtrer par priorité journalctl -p err # Erreurs
uniquement journalctl -p warning # Avertissements et plus # Nettoyer les logs sudo journalctl --vacuum-
time=30d # Garder 30 jours sudo journalctl --vacuum-size=100M # Limite de taille
```

2.4 Gestion du stockage

2.4.1 Surveillance de l'espace disque

```
# Espace disque global df -h # Lisible par l'homme df -i # Utilisation des inodes # Utilisation par répertoire
du -sh /home/* # Taille de chaque utilisateur du -sh /* 2>/dev/null | sort -hr # Plus gros répertoires # Trouver
les gros fichiers find /home -type f -size +100M # Fichiers > 100MB find /var -type f -size +1G -exec ls -lh {}
\; # Fichiers > 1GB avec détails # Analyse avancée ncd /home # Interface interactive pour analyser
l'espace
```

2.4.2 Gestion des partitions

```
# Lister les disques et partitions lsblk # Arbre des périphériques fdisk -l # Informations détaillées (root) blkid
# UUID et systèmes de fichiers # Informations sur les montages mount # Points de montage actuels cat
/proc/mounts # Vue du noyau findmnt # Affichage en arbre # Monter/démonter manuellement sudo mount
/dev/sdb1 /mnt/usb sudo umount /mnt/usb sudo mount -t ext4 -o ro /dev/sdb1 /mnt/readonly # Options
spécifiques
```

2.4.3 Configuration du montage automatique

```
# Fichier /etc/fstab - configuration permanente # Format: device mountpoint filesystem options dump pass #
Exemple d'entrées fstab: UUID=12345678-1234-1234-1234-123456789012 /home ext4 defaults 0 2
/dev/sdb1 /mnt/data ext4 defaults,noatime 0 0 //server/share /mnt/samba cifs
username=user,password=pass 0 0 # Tester la configuration fstab sudo mount -a # Monter tout ce qui est
dans fstab sudo findmnt --verify # Vérifier la syntaxe fstab
```

2.5 Gestion réseau

2.5.1 Configuration réseau

```
# Informations réseau modernes (ip tools) ip addr show # Adresses IP ip route show # Table de routage ip
link show # Interfaces réseau # Configuration temporaire sudo ip addr add 192.168.1.100/24 dev eth0 sudo
ip route add default via 192.168.1.1 sudo ip link set eth0 up # Anciens outils (toujours utilisés) ifconfig #
Informations interfaces route -n # Table de routage netstat -rn # Table de routage
```

2.5.2 Configuration réseau persistante

Ubuntu/Debian avec Netplan :

```
# Fichier /etc/netplan/01-config.yaml network: version: 2 renderer: networkd ethernets: enp0s3: dhcp4: true
enp0s8: addresses: - 192.168.1.100/24 gateway4: 192.168.1.1 nameservers: addresses: [8.8.8.8, 1.1.1.1] #
Appliquer la configuration sudo netplan try # Test (revient automatiquement) sudo netplan apply #
Application définitive
```

CentOS/RHEL :

```
# Fichier /etc/sysconfig/network-scripts/ifcfg-eth0 TYPE=Ethernet BOOTPROTO=static NAME=eth0
DEVICE=eth0 ONBOOT=yes IPADDR=192.168.1.100 NETMASK=255.255.255.0 GATEWAY=192.168.1.1
DNS1=8.8.8.8 DNS2=1.1.1.1 # Redémarrer le réseau sudo systemctl restart NetworkManager
```

2.5.3 Outils de diagnostic réseau

```
# Test de connectivité ping -c 4 google.com # Test de base ping6 -c 4 google.com # IPv6 traceroute
google.com # Chemin réseau mtr google.com # Traceroute continu # Test des ports nmap localhost #
Scanner de ports nc -zv google.com 80 # Test de port spécifique telnet google.com 80 # Connexion
manuelle # Surveillance du trafic sudo netstat -tulpn # Ports en écoute avec processus sudo ss -tulpn #
Version moderne sudo iftop # Monitoring trafic par interface sudo nethogs # Trafic par processus
```

2.6 Gestion des logs système

2.6.1 Organisation des logs

Fichier/Répertoire	Contenu	Importance
/var/log/syslog	Messages système généraux	Critique
/var/log/auth.log	Authentifications et autorisations	Sécurité
/var/log/kern.log	Messages du noyau	Matériel/drivers
/var/log/apache2/	Logs du serveur web Apache	Applications
/var/log/mysql/	Logs de la base de données	Applications

2.6.2 Rotation des logs avec logrotate

```
# Configuration principale cat /etc/logrotate.conf # Exemple de configuration pour Apache # Fichier
/etc/logrotate.d/apache2 /var/log/apache2/*.log { weekly missingok rotate 52 compress delaycompress
notifempty create 644 root adm postrotate systemctl reload apache2 endscrip } # Test de la rotation sudo
logrotate -d /etc/logrotate.conf # Mode debug sudo logrotate -f /etc/logrotate.conf # Forcer la rotation
```

2.6.3 Surveillance et analyse des logs

```
# Surveillance en temps réel tail -f /var/log/syslog multitail /var/log/syslog /var/log/auth.log # Plusieurs
fichiers # Recherche dans les logs grep "ERROR" /var/log/apache2/error.log grep -i "failed" /var/log/auth.log
| tail -10 zgrep "pattern" /var/log/syslog.*.gz # Dans les logs compressés # Analyse statistique awk '{print
$12}' /var/log/apache2/access.log | sort | uniq -c | sort -nr # Analyse des codes de réponse HTTP # Outils
avancés sudo fail2ban-client status # Protection contre les attaques sudo goaccess
/var/log/apache2/access.log --log-format=COMBINED # Analyse interactive des logs web
```

2.7 Sauvegardes et archivage

2.7.1 Création d'archives avec tar

```
# Créer une archive tar -czf sauvegarde.tar.gz /home/alice/ # Compression gzip tar -cjf sauvegarde.tar.bz2
/home/alice/ # Compression bzip2 tar -cJf sauvegarde.tar.xz /home/alice/ # Compression xz # Options
courantes: # c = créer # z = gzip # j = bzip2 # J = xz # f = fichier # v = verbose # t = lister le contenu # x =
extraire # Lister le contenu d'une archive tar -tzf archive.tar.gz # Extraire une archive tar -xzf archive.tar.gz -
C /destination/ tar -xzf archive.tar.gz fichier_specifique
```

2.7.2 Synchronisation avec rsync

```
# Synchronisation locale rsync -av /source/ /destination/ # Options importantes: # -a = archive (préserve
permissions, dates, etc.) # -v = verbose # -n = dry-run (test) # -z = compression # --delete = supprimer les
fichiers qui n'existent plus dans la source # --exclude = exclure des fichiers/répertoires # Synchronisation
distante rsync -avz /local/dir/ user@server:/remote/dir/ rsync -avz --delete user@server:/remote/dir/
/local/backup/ # Sauvegarde incrémentale avec liens durs rsync -av --link-dest=/backup/previous/ /home/
/backup/current/
```

2.7.3 Scripts de sauvegarde automatisée

```
#!/bin/bash # Script de sauvegarde automatique DATE=$(date +%Y%m%d_%H%M%S) SOURCE="/home"
DEST="/backup/daily" LOG="/var/log/backup.log" echo "$(date): Début de la sauvegarde" >> $LOG # Créer
le répertoire de destination mkdir -p "$DEST" # Sauvegarde avec rsync rsync -av --delete \ --
exclude='*.tmp' \ --exclude='Cache/' \ "$SOURCE/" "$DEST/" >> $LOG 2>&1 # Créer une archive
hebdomadaire le dimanche if [ $(date +%u) -eq 7 ]; then tar -czf "/backup/weekly/backup_$DATE.tar.gz" -C
"$DEST" . >> $LOG 2>&1 fi echo "$(date): Sauvegarde terminée" >> $LOG
```

3. Travaux pratiques et exercices

3.1 TP1 : Installation et découverte du système

Exercice 1.1 : Exploration du système

Durée : 30 minutes

Objectifs :

- Se familiariser avec l'arborescence Linux
- Comprendre les informations système
- Utiliser les commandes de base

Consignes :

1. Identifiez votre distribution Linux et sa version

```
lsb_release -a  
uname -a
```

2. Explorez l'arborescence racine et documentez le rôle de 5 répertoires
3. Trouvez votre répertoire personnel et listez son contenu avec permissions
4. Vérifiez l'espace disque disponible sur toutes les partitions
5. Listez les 10 dernières connexions au système

Livrables : Document avec captures d'écran et explications

Exercice 1.2 : Manipulation de fichiers

Durée : 45 minutes

Scenario : Vous devez organiser les documents d'un projet

1. Créez l'arborescence suivante dans votre home :

```
projet_gl2/ |—— docs/ | |—— cahier_charges.txt | |—— specifications/ |—— src/ |  
|—— main.c | |—— utils.c |—— tests/ |—— test_main.c
```

2. Remplissez chaque fichier .txt avec du contenu approprié
3. Définissez les permissions suivantes :
 - Répertoire projet_gl2 : rwx pour vous, r-x pour le groupe, rien pour les autres
 - Fichiers .c : rw- pour vous, r-- pour le groupe et les autres
 - Fichiers .txt : rw- pour vous et le groupe, r-- pour les autres
4. Créez une archive compressée de tout le projet
5. Créez un script qui compte les lignes de tous les fichiers .c

3.2 TP2 : Gestion des utilisateurs et groupes

Exercice 2.1 : Administration des utilisateurs

Durée : 60 minutes

Scenario : Vous administrez un système pour une équipe de développement

1. Créez les groupes suivants :
 - developers : pour les développeurs
 - testers : pour les testeurs
 - managers : pour les chefs de projet
2. Créez les utilisateurs suivants :

Utilisateur	Groupe principal	Groupes supplémentaires	Shell
alice	developers	managers	/bin/bash
bob	developers	-	/bin/bash
charlie	testers	-	/bin/bash

3. Configurez les politiques de mots de passe :
 - Expiration tous les 90 jours
 - Avertissement 7 jours avant expiration
 - Compte verrouillé après 30 jours d'inactivité
4. Créez un répertoire partagé /shared/project avec les permissions appropriées
5. Testez les connexions et permissions avec chaque utilisateur

3.3 TP3 : Administration système

Exercice 3.1 : Surveillance du système

Durée : 45 minutes

1. Installez et configurez les outils de surveillance :

```
sudo apt update  
sudo apt install htop iotop nethogs
```

2. Créez un script de surveillance qui affiche :
 - Charge système (load average)
 - Utilisation mémoire et swap
 - Espace disque utilisé
 - Processus consommant le plus de CPU
 - Connexions réseau actives
3. Configurez ce script pour s'exécuter toutes les 5 minutes et logger les résultats
4. Créez des alertes quand :
 - L'utilisation disque dépasse 80%
 - La charge système dépasse 2.0
 - L'utilisation mémoire dépasse 90%

Exercice 3.2 : Gestion des services

Durée : 60 minutes

Scénario : Déploiement d'un serveur web

1. Installez Apache2 et configurez-le :

```
sudo apt install apache2
```

2. Créez un site web personnalisé dans /var/www/monsite/
3. Configurez un Virtual Host pour ce site
4. Activez les modules nécessaires (rewrite, ssl)
5. Configurez les logs personnalisés
6. Mettez en place la rotation des logs
7. Créez un script de sauvegarde automatique du site
8. Testez l'arrêt/redémarrage du service
9. Surveillez les logs en temps réel lors de tests

3.4 TP4 : Sécurité et réseau

Exercice 4.1 : Sécurisation SSH

Durée : 45 minutes

1. Modifiez la configuration SSH (/etc/ssh/sshd_config) :
 - Changez le port par défaut (2222)
 - Désactivez la connexion root directe
 - Activez l'authentification par clés
 - Limitez les utilisateurs autorisés
2. Générez des paires de clés SSH pour vos utilisateurs
3. Configurez l'authentification sans mot de passe
4. Installez et configurez fail2ban pour protéger SSH
5. Testez les connexions et surveillez les tentatives d'intrusion

Exercice 4.2 : Script d'administration personnalisé

Durée : 90 minutes

Projet final : Créez un outil d'administration système

Fonctionnalités requises :

1. Menu interactif avec les options :
 - Affichage des informations système
 - Gestion des utilisateurs (création, suppression, modification)
 - Surveillance des processus
 - Gestion des services
 - Sauvegarde système
 - Analyse des logs
2. Gestion d'erreurs et validation des entrées
3. Logging des actions effectuées
4. Interface colorée et intuitive

5. Aide contextuelle

Exemple de structure :

```
#!/bin/bash # Couleurs pour l'interface RED='\033[0;31m' GREEN='\033[0;32m' YELLOW='\033[1;33m'
NC='\033[0m' # No Color # Fonction d'affichage du menu show_menu() { clear echo -e "${GREEN}===
Administration System Tool ===${NC}" echo "1. System Information" echo "2. User Management" echo
"3. Process Monitoring" echo "4. Service Management" echo "5. System Backup" echo "6. Log
Analysis" echo "7. Exit" echo -n "Choose an option [1-7]: " } # Fonction principale main() { while true; do
show_menu read choice case $choice in 1) show_system_info ;; 2) user_management ;; 3)
process_monitoring ;; 4) service_management ;; 5) system_backup ;; 6) log_analysis ;; 7) exit 0 ;; *)
echo -e "${RED}Invalid option${NC}" ;; esac read -p "Press Enter to continue..." done } main
```

3.5 Évaluation et critères

3.5.1 Grille d'évaluation

Compétence	Critères	Points
Navigation système	Maîtrise des commandes de base, arborescence	/20
Gestion fichiers	Manipulation, permissions, archivage	/20
Administration utilisateurs	Création, modification, sécurité	/20
Gestion processus/services	Surveillance, contrôle, optimisation	/20
Scripting	Automatisation, gestion d'erreurs	/20

3.5.2 Ressources complémentaires

Documentation :

- Pages man : man commande
- Info pages : info commande
- Help intégré : commande --help
- Documentation en ligne : /usr/share/doc/

Sites web utiles :

- Linux Documentation Project : tldp.org
- Ubuntu Documentation : help.ubuntu.com
- Red Hat Documentation : access.redhat.com
- Stack Overflow : stackoverflow.com (tag linux)

Livres recommandés :

- "Linux Administration Handbook" - Nemeth, Snyder, Hein
- "The Linux Command Line" - William Shotts
- "Unix and Linux System Administration Handbook" - Nemeth et al.

Points clés à retenir

- Linux est un système puissant qui nécessite de la pratique
- La ligne de commande est plus efficace que l'interface graphique pour l'administration
- Toujours tester les commandes dans un environnement sûr
- Documenter et commenter vos scripts
- La sécurité doit être une préoccupation constante
- Les sauvegardes sont critiques avant toute modification importante

Support de cours - Administration des systèmes et réseaux Linux (GL2)

Année académique 2025-2026 - Volume horaire : 30h (CM, TD, TP, TPE)

Document réalisé le :22/09/2025