

# Maîtrisez le système d'exploitation Linux

Plutôt initiation au système  
Linux

**02 au 05  
2025**





# Presentation

## Qui vous êtes?



# 1

# Introduction à Linux

## Définition

**Linux** est un système d'exploitation (OS) **libre et open source** basée sur les principes UNIX. Il permet à un ordinateur d'exécuter des **programmes**, de gérer les **fichiers**, la **mémoire**, les **utilisateurs**, les **périphériques**, et de communiquer avec le **matériel**.

Linux est composé de deux parties principales :

**Le noyau (Kernel)** : cœur du système, gère le matériel.

**Les logiciels du système (GNU)** : outils essentiels pour travailler (bash, ls).

**GNU/Linux** : association du noyau Linux + outils GNU

Principalement utilisés dans les serveurs (90% des serveurs web), le cloud, DevOps, cybersécurité, Android (noyau Linux).

# Historique

- **Origine d'UNIX**

**1969** : AT&T Bell Labs crée **UNIX**.

Objectif : fournir un système multitâche, multi-utilisateur, stable.

UNIX devient rapidement la base de nombreux systèmes modernes.

- **Apparition du projet GNU**

**1983** : Richard Stallman lance le projet **GNU**

Fournir un système complet, libre et compatible UNIX.

GNU crée des outils essentiels : bash, gcc, ls, cp, etc.

Mais il manquait encore un **noyau libre**.

- **Naissance de Linux**

**1991**, Linus Torvalds, étudiant finlandais, développe un noyau libre : **Linux**.

Il publie la première version (0.01) en invitant la communauté à contribuer.

- **Fusion GNU + Linux**

Les outils GNU + le noyau Linux = **GNU/Linux**, le système utilisé aujourd'hui

# Les différentes distributions Linux

Une **distribution (distro)** = noyau Linux + outils + interface + gestionnaire de paquets.

Les plus connues :

## **Debian**

Très stable, utilisée pour serveurs.

Base de plusieurs distros (Ubuntu, Kali).

Gestionnaire : apt.

## **Ubuntu**

Basée sur Debian, plus user-friendly.

Idéal pour débutants, ordinateurs personnels, serveurs.

Versions : Desktop, Server, LTS.

## **Kali Linux**

Distro spécialisée en cybersécurité & tests d'intrusion.

Inclus nmap, metasploit, burp suite, hydra, etc.

# Les différentes distributions Linux

## **CentOS / RHEL**

Basées sur Red Hat Enterprise Linux.

Très utilisées dans les entreprises et serveurs professionnels.

Gestionnaire : yum / dnf

## **Fedora**

Innovante, version communautaire de Red Hat.

Pour développeurs.

## **Autres**

Arch Linux (minimaliste),

Linux Mint (facile pour débutants),

OpenSUSE (stable et pro),

Parrot OS (cybersécurité).

# Différences entre Linux et Windows

Critère	Linux	Windows
Licence	Open source, gratuit	Propriétaire, payant
Sécurité	Très sécurisé, peu de virus	Plus vulnérable aux malwares
Stabilité	Très stable, utilisé sur serveurs	Peut devenir lent ou instable
Personnalisation	Très flexible, modifiable	Peu personnalisable
Mode d'utilisation	Terminal + interface	Principalement interface graphique
Performances	Très léger, idéal pour vieux PC	Plus lourd, nécessite ressources
Applications	Serveurs, dev, scripts	Jeux, bureautique grand public
Gestion des fichiers	Arborescence UNIX standard	Structure Windows avec C:\

**Linux** : sécurisé, stable, puissant, idéal pour serveurs, dev, cybersécurité

**Windows** : plus orienté utilisateurs classiques et jeux

# Le noyau, Shell et environnement graphique

## Noyau (Kernel)

Cœur du système. Qui communique entre : logiciel - le matériel. Il gère la mémoire, les processus, les pilotes de périphériques, le système de fichiers, la sécurité.

## Le Shell

C'est Interpréteur de commandes. Il permet de dialoguer avec le système via le terminal.

Principaux Shells : **Bash (le plus utilisé)** , Zsh , Fish, Sh

## Environnement graphique (GUI)

Interface visuelle (bureau, fenêtres, icônes).

Elle repose sur le serveur graphique ( **Xorg** ou **Wayland**).

Environnements graphiques : GNOME, KDE Plasma, XFCE, LXDE, Cinnamon

# Architecture d'un système GNU/Linux

la structure générale du système :

## 1. Le noyau (kernel)

Le cœur du système, gère le matériel.

## 2. Bibliothèques système

Fournissent des fonctionnalités aux programmes.

## 3. Les utilitaires GNU

Commandes de base : ls, cp, mv, rm, mkdir, etc.

## 4. Le Shell

Interface en ligne de commande.

## 5. Serveur d'affichage

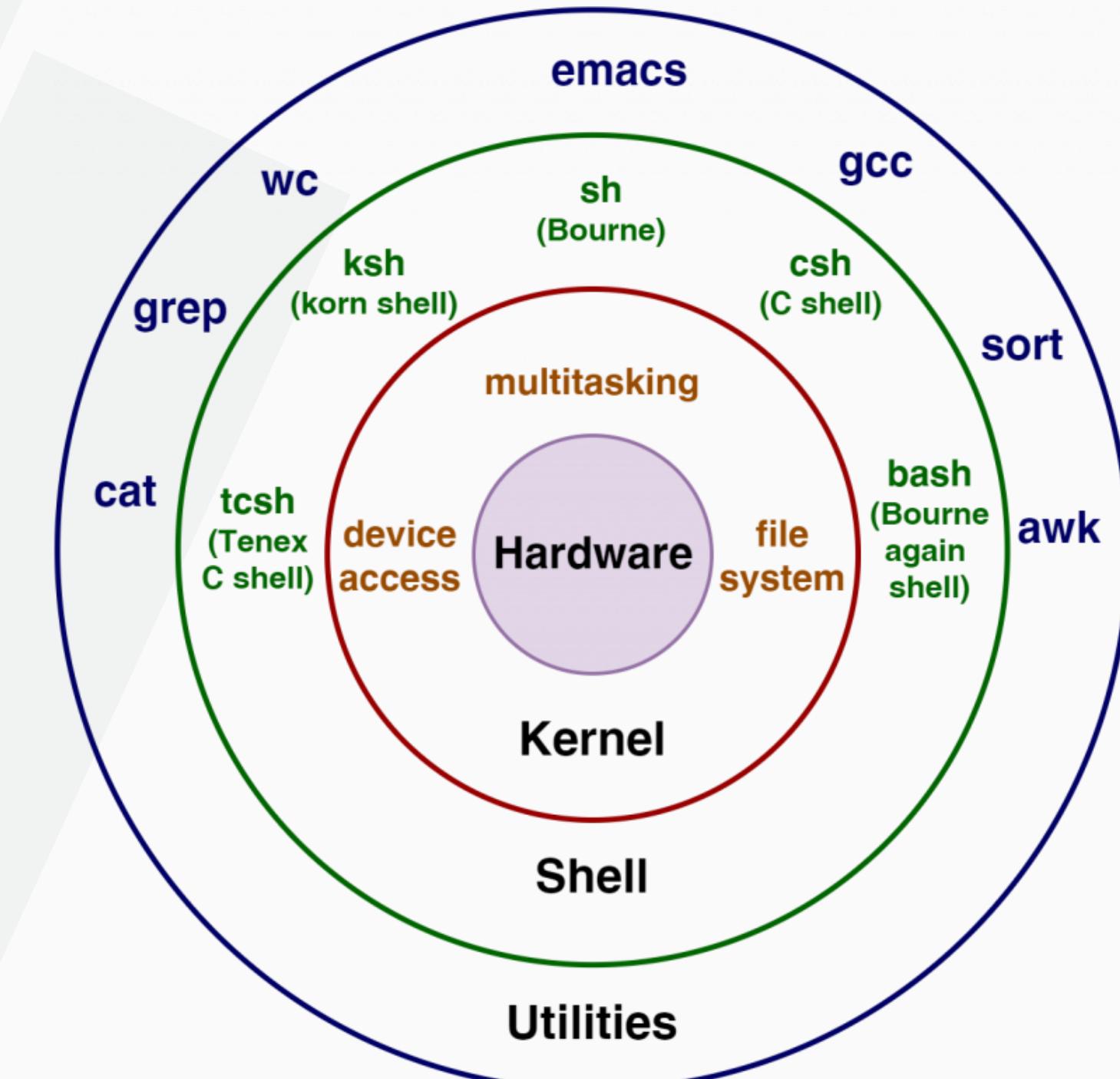
Xorg ou Wayland (permet l'affichage graphique).

## 6. Environnement graphique

GNOME, KDE...

## 7. Applications

Navigateur web, éditeur de texte, outils de gestion...



## 2

# Prise en main du Terminal

## Commandes de base

### Lister les fichiers

Affiche le contenu d'un répertoire.

ls : liste les fichiers

ls -l : mode détaillé

ls -a : afficher les fichiers cachés

ls -lh : formats lisibles (Ko, Mo)

### Changer de répertoire

cd dossier : entrer dans *dossier*

cd .. : remonter d'un niveau

cd /home/user : chemin complet

cd : revenir au dossier personnel

cd - : revenir au dossier précédent

pwd : Afficher le répertoire courant

clear : Effacer l'écran

### Manipulation des fichiers et dossiers

cp — Copier un fichier ou dossier

mv — Déplacer ou renommer

rm — Supprimer fichiers ou dossiers

mkdir — Créer un dossier

rmdir — Supprimer un dossier vide

### Lecture et édition de fichiers

cat — Afficher le contenu d'un fichier

less / more — Lire un fichier page par page

nano — Éditeur de texte simple

vim — Éditeur avancé

# Commandes de base

## Chemins absolus

Toujours depuis la racine /

/home/amoros/Documents/fichier.txt  
/etc/hosts

## Chemin relatif

Basé sur votre position actuelle.

Documents/fichier.txt  
..../images/photo.jpg

## Auto-complétion

Avec la touche **TAB** :

Complète automatiquement  
les fichiers, dossiers, commandes  
Evite les fautes de frappe

## Historique des commandes

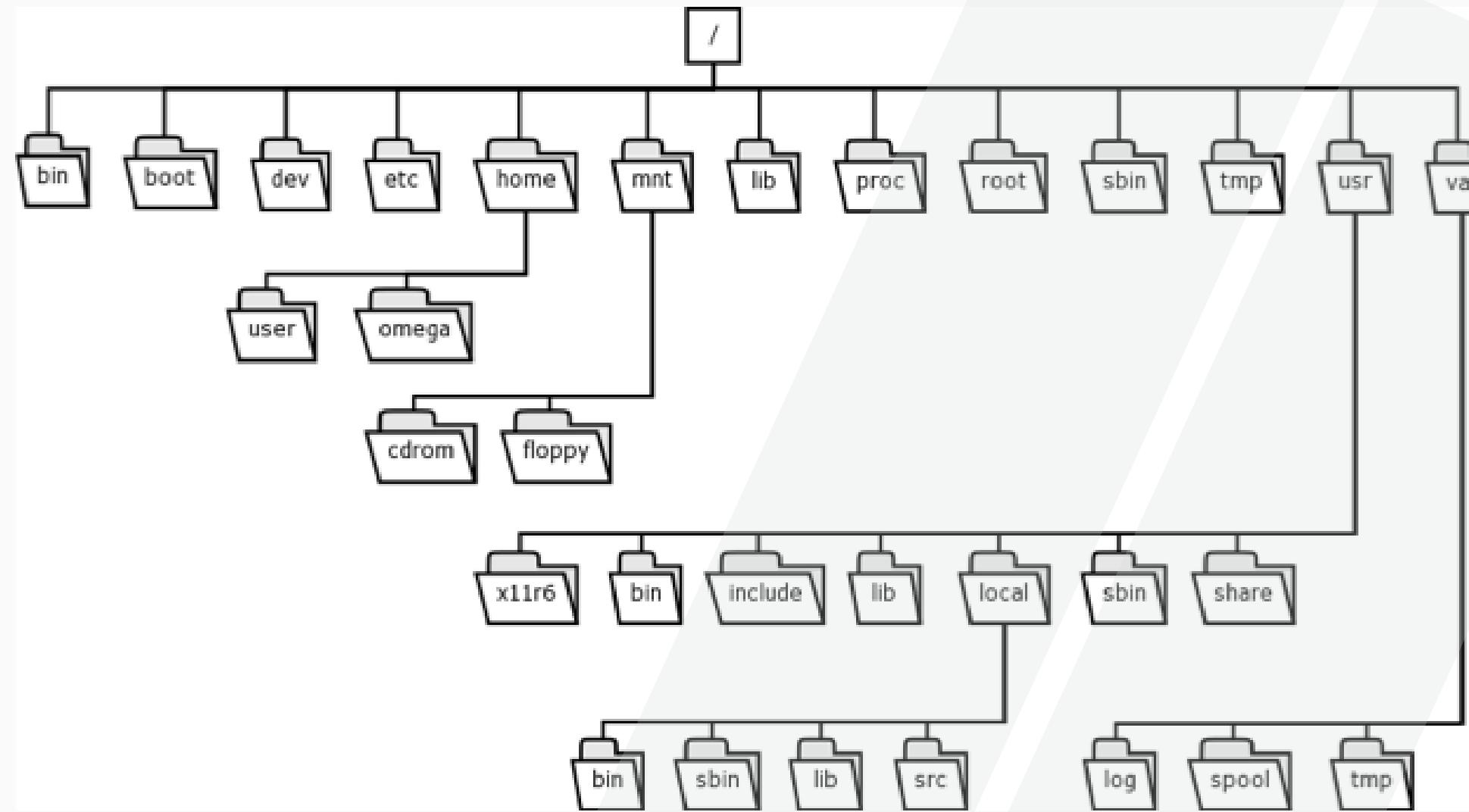
Afficher l'historique : **history**

↑ : remonter dans l'historique

!! : répéter la dernière commande

# Gestion des fichiers et du système de fichiers

## Arborescence Linux



## Droits et permissions :

lecture (r), écriture (w), exécution (x)

Utilisateurs u / groupes g / autres o

modifier les droits : chmod, chown,

chgrp

# Gestion des utilisateurs et groupes

## utilisateurs et groupes

Ajouter, supprimer, modifier un utilisateur  
(`useradd`, `usermod`, `userdel`)

Gestion des mots de passe (`passwd`)

Groupes (`groupadd`, `groupdel`, `groups`)

Fichiers importants : `/etc/passwd`, `/etc/group`,  
`/etc/shadow`

Notion de superutilisateur (root) et commande `sudo`

## Gestion des processus

Afficher les processus (`ps`, `top`, `htop`, `pgrep`)

Gérer les processus (`kill`, `pkill`, `nice`, `renice`)

Services système :

démarrer / arrêter / redémarrer (`systemctl`  
`start|stop|restart`)

activer au démarrage (`systemctl enable`)

vérifier l'état (`systemctl status`)

## Gestion des logiciels

Gestionnaire de paquets selon distribution :

**Debian/Ubuntu** : `apt`, `dpkg`

**RedHat/CentOS** : `yum`, `dnf`, `rpm`

Installer et désinstaller un logiciel, Rechercher un paquet, Mise à jour du système, Ajouter un dépôt (repository)

# Réseau sous Linux

## Réseau

Vérifier l'état réseau : `ip`, `ifconfig`, `iwconfig`

Gestion des interfaces

Configuration d'une adresse IP

Manipuler les ports : `Ping`, `traceroute`, `nslookup`, `dig`  
`netstat`, `ss`, `nmap`

Gestion des firewall : `ufw`, `iptables`, `firewalld`

## Gestion du stockage

Disques et partitions (`lsblk`, `fdisk`, `parted`)

Systèmes de fichiers : `ext4`, `xfs`, `fat32`, `ntfs...`

Monter / démonter un volume (`mount`,  
`umount`)

Vérification du système (`df`, `du`, `fsck`)

# 3

# Automatisation et Scripts Shell

## Introduction au Shell Script

Un **Shell Script** est un fichier texte contenant une suite de commandes Linux, exécutées dans l'ordre.

C'est l'équivalent d'un petit programme qui automatise des tâches.

Le shell le plus utilisé : **Bash** : `/bin/bash`

Chaque script commence généralement par la ligne : `#!/bin/bash`

# Variables, conditions, boucles

## Variables

```
nom="ODC"  
echo "Bonjour $nom "
```

## Variables système :

```
echo $USER  
echo $HOME  
echo $PATH
```

## Boucle for

```
for i in 1 2 3 4 5; do  
    echo "Nombre : $i "
```

Done

## Boucle While

```
x=1  
while [ $x -le 5 ]; do  
    echo "Valeur : $x"  x=$((x+1))  
done
```

## Conditions (if / else)

```
if [ $age -ge 18 ]; then  
    echo "Majeur"  
else  
    echo "Mineur"  
fi
```

## Tester si un fichier existe

```
if [ -f fichier.txt ]; then  
    echo "Le fichier existe"  
fi
```

## Tester si un dossier existe

```
if [ -d /home/amoros ]; then  
    echo "Dossier trouvé"  
fi
```

# Créer un script (.sh) et le rendre exécutable

## Étape 1 : créer le fichier

```
nano backup.sh
```

## Étape 2 : ajouter du code

```
#!/bin/bash
```

```
echo "Sauvegarde en cours..."
```

```
cp -r /home/user/Documents /home/user/Backup
```

```
echo "Terminé !"
```

```
Ctrl +O +Entré +Ctrl + X
```

## Étape 3 : rendre le script exécutable

```
chmod +x backup.sh
```

## Étape 4 : exécuter le script

```
./backup.sh
```

# Crontab : automatiser des tâches

Le programme **cron** permet d'exécuter des scripts automatiquement à une heure ou une fréquence précise.

## Ouvrir la crontab

crontab -e

M(0-59) H(0-23) J(1-31) M(1-12) J(0-7) commande/script

## Exécuter un script chaque jour à 23h

0 23 \* \* \* /home/user/backup.sh

## Nettoyer un dossier toutes les 10 minutes

\*/10 \* \* \* \* rm -rf /tmp/\*

## Envoyer un log chaque matin

0 7 \* \* \* echo "Vérification OK" >> /var/log/rapport.log

# Journalisation/logger

## **Journalisation : logger et /var/log**

**logger** permet d'envoyer des messages dans les logs du système.

Exemple :

**logger "Sauvegarde terminée avec succès"**

Ce message apparaît ensuite dans :

**/var/log/syslog (Debian/Ubuntu)**

Créer un log personnalisé dans un script

**#!/bin/bash**

**logger "Début du script de maintenance"**

**echo "Nettoyage..." >> /var/log/maintenance.log**

**rm -rf /tmp/\*logger "Fin du script de maintenance "**

## **Consulter les logs**

Pour lire la fin du fichier : **tail -f /var/log/syslog**

Pour rechercher une ligne : **grep "erreur" /var/log/syslog**

**4**

## **Conclusion**

Revoyez dès le début le contenu de formation, ce ça  
le resumé. 😊



orangedigitalcenter.fianar@gmail.com

---