

Matière : Logiciels libres et open source : Cours 03

1. Introduction aux Environnements Libres

Dans le monde de l'informatique moderne, les **environnements libres** occupent une place essentielle. Ils incarnent une philosophie basée sur le **partage, la collaboration et la liberté d'utilisation** des logiciels. Contrairement aux environnements propriétaires, qui limitent l'accès au code source et restreignent les droits d'utilisation, les environnements libres offrent aux utilisateurs la possibilité d'**utiliser, d'étudier, de modifier et de redistribuer** les logiciels selon leurs besoins.

Les environnements libres ne se résument pas à des programmes isolés, mais forment un **écosystème complet** regroupant :

- Un **système d'exploitation libre**, comme **GNU/Linux** ;
- Des **applications libres** (bureautique, multimédia, développement, Internet, etc.) ;
- Et une **communauté active** d'utilisateurs et de développeurs qui assurent le maintien, la documentation et l'évolution de ces outils.

L'un des plus grands succès de cette approche est le **projet GNU/Linux**, né de la rencontre entre le projet **GNU** (initié par Richard Stallman en 1983) et le noyau **Linux** (créé par Linus Torvalds en 1991). Ce système d'exploitation libre est aujourd'hui utilisé partout : dans les serveurs, les supercalculateurs, les téléphones Android, et même dans l'éducation.

Ainsi, étudier les environnements libres, c'est comprendre :

- **Une autre manière de concevoir l'informatique**, fondée sur la liberté et la transparence ;
- **Des outils puissants et sécurisés**, largement utilisés dans le monde professionnel ;
- Et **une culture du partage des connaissances**, essentielle à la science et à l'innovation.

2. Présentation du système GNU/Linux

2.1. Qu'est-ce que le mouvement GNU ?

En 1984, Richard Matthew Stallman, chercheur en informatique du MIT quitte son poste et se consacre à l'écriture d'un système d'exploitation Libre du nom de GNU . Il annonce l'année suivante la création de la FSF afin de supporter ce projet.

C'est durant ces années qu'il écrit ce qui deviendra les préceptes du Logiciel Libre. La concrétisation en est la publication en 1989 de la première version de la licence GPL qui sera alors le fondement éthique, juridique et politique du mouvement du Libre.

2.2. Linux, juste le noyau !

«Au sens strict, Linux est le nom du noyau de système d'exploitation libre, multitâche, multiplate-forme et multi-utilisateur de type UNIX créé par Linus Torvalds, souvent désigné comme le noyau Linux.»

Le projet GNU arrive en 1991 avec de très nombreux outils libres, mais il lui manque un élément central : le noyau. Cet élément est essentiel car il gère la mémoire, le microprocesseur, les périphériques comme le clavier, la souris, les disques durs. . .

C'est à cette époque qu'un étudiant finlandais, Linus Torvalds, commence à développer un noyau et demande aux personnes intéressées d'y contribuer. La licence GPL a été publiée à la même époque et Linus Torvalds s'est laissé persuader de placer son noyau sous cette dernière.

Le système d'exploitation actuellement connu est donc un assemblage des outils GNU fonctionnant sur un noyau Linux, on parle donc de GNU/Linux avec le slash, « / » pour « GNU sur Linux ».

Définition

GNU/Linux est un système d'exploitation complètement Libre et performant. Il est hautement configurable. Il ne dépend pas d'une multinationale. Il est supporté par une grande communauté d'utilisateurs souvent prêts à vous aider. Quelque soit votre domaine de compétence, vous pouvez participer à l'amélioration de GNU/Linux pour que ce dernier évolue dans votre intérêt. Ce n'est pas un simple logiciel gratuit, mais un Logiciel Libre. Ce qui garantit qu'il restera accessible et gratuit pour tous, sans discrimination.

Linux est un système d'exploitation **hybride**, car il permet de travailler aussi bien à travers une **interface graphique conviviale (GUI)** qu'au moyen du **terminal en ligne de commande (CLI)**. Cette double approche offre une grande flexibilité : l'environnement graphique facilite l'utilisation

quotidienne pour les débutants et les tâches courantes, tandis que le terminal donne accès à la **pleine puissance du système**. En effet, les **commandes Linux** permettent d'automatiser les opérations, de gérer efficacement les fichiers, les utilisateurs ou les processus, et d'administrer le système à distance. Dans le monde professionnel, notamment pour les administrateurs systèmes et les développeurs, la maîtrise des commandes est considérée comme **indispensable**, car elle offre un **contrôle précis, rapide et léger** sur le fonctionnement du système, sans dépendre d'une interface graphique.

2.3. Un système d'exploitation multiplateforme

Linux est multiplateforme.

Ce terme veut tout simplement dire que Linux est disponible sur plusieurs types de machines ou architecture de processeur. Ainsi, on va pouvoir trouver Linux sur une machine de type P.C. tel que vous connaissez et que vous utilisez certainement, mais aussi sur les Macintosh, les Smartphones ou encore sur les super calculateurs.

3. Principales distributions (Ubuntu, Fedora, Debian)

C. Qu'est ce qu'une distribution ?

Une distribution Linux est un ensemble qui se compose d'un noyau Linux et d'une sélection d'applications qui sont maintenues par une entreprise ou une communauté d'utilisateurs.

L'objectif d'une distribution est d'optimiser le noyau et les applications qui sont exécutées sur le système d'exploitation pour un certain usage ou groupe d'utilisateurs.

Les distributions comprennent souvent des outils qui leur sont spécifiques pour l'installation de logiciels et l'administration du système. C'est pourquoi certaines distributions sont principalement utilisées pour des environnements de bureau où elles doivent être faciles à utiliser, tandis que d'autres sont principalement utilisées pour fonctionner sur des serveurs afin d'utiliser les ressources disponibles de la manière la plus performante possible.

En fait, chaque distribution a sa cible : certaines sont orientées sur la facilité d'utilisation, d'autres sont pour les véritables « geeks », certaines sont spécialisées pour l'utilisation dans le domaine scolaire ou musical, d'autres encore se veulent très légères et fonctionner sur des PC à faible performance.

Une autre façon de classer les distributions est de se référer à la famille de distribution à laquelle elles appartiennent (ci-dessous quelques exemples des distributions Linux) :

- La distribution **Debian GNU/Linux** est la plus grande distribution de la famille des distributions Debian. Le projet Debian GNU/Linux a été lancé par Ian Murdock en 1993. Aujourd'hui, des milliers de bénévoles travaillent sur le projet. Debian GNU/Linux vise à fournir un système d'exploitation très fiable. Elle promeut également la vision de Richard Stallman d'un système d'exploitation qui respecte les libertés de l'utilisateur d'exécuter, d'étudier, de distribuer et d'améliorer les logiciels. C'est pourquoi elle ne fournit aucun logiciel propriétaire par défaut.
- **Ubuntu** est une autre distribution basée sur Debian qui mérite d'être mentionnée. Ubuntu a été créée par Mark Shuttleworth et son équipe en 2004, avec pour mission d'apporter un environnement de bureau Linux facile à utiliser. La mission d'Ubuntu est de fournir des logiciels libres à tous dans le monde entier ainsi que de réduire le coût des services professionnels. La distribution a une sortie prévue tous les six mois, avec une sortie à support à long terme tous les deux ans.
- **Red Hat** est une distribution Linux développée et maintenue par l'entreprise de logiciels du même nom, qui a été rachetée par IBM en 2019. La distribution Red Hat Linux en 1994 et renommée en 2003 Red Hat Enterprise Linux, souvent abrégée en RHEL. Elle est offerte aux entreprises en tant que solution d'entreprise fiable, son soutien est fourni par Red Hat et inclus des logiciels visant à faciliter l'utilisation de Linux dans les environnements de serveurs professionnels. Certains de ses composants nécessitent des abonnements ou des licences payantes.
- Le projet **CentOS** utilise le code source librement disponible de Red Hat Enterprise Linux et le compile en une distribution qui est disponible entièrement gratuitement, mais en retour n'est accompagnée d'aucun support commercial.
- Le projet **Fedora** a été fondé en 2003 et crée une distribution Linux destinée aux ordinateurs de bureau.

Des distributions indépendantes ont été publiées au fil des années. Certaines d'entre elles sont basées sur Red Hat ou Ubuntu, d'autres sont conçues pour améliorer une propriété spécifique d'un système ou d'un matériel. Il existe des distributions construites avec des fonctionnalités spécifiques comme **QubesOS**, un environnement de bureau très sécurisé, ou **Kali Linux**, qui fournit un environnement permettant d'exploiter les vulnérabilités des logiciels, principalement utilisé par les testeurs de pénétration.

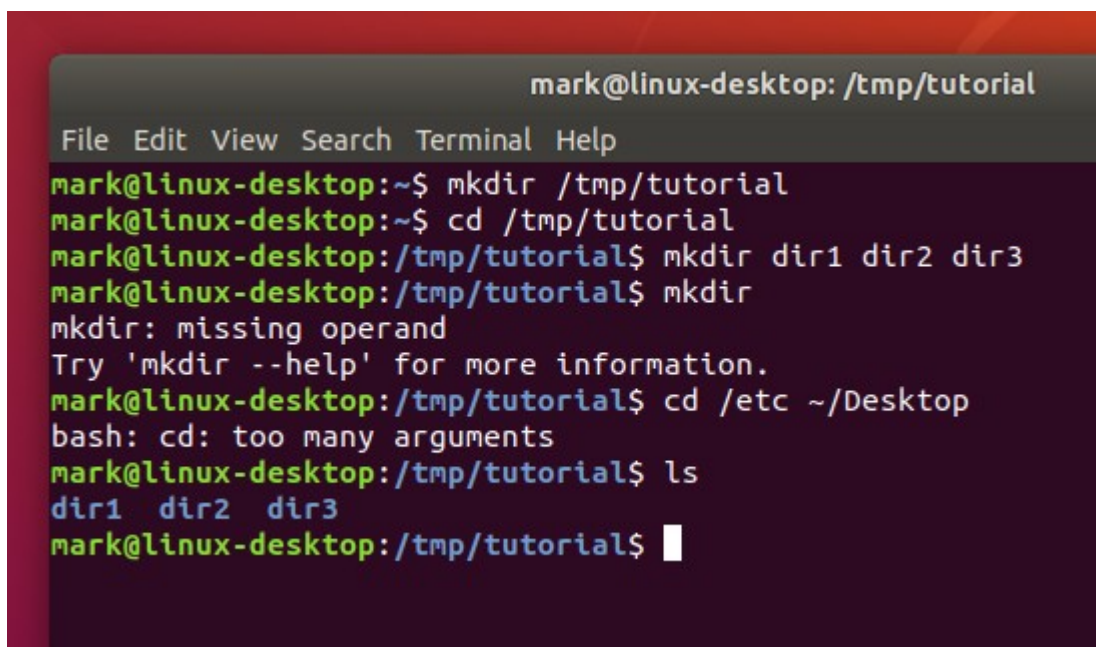
Il existe également des distributions conçues spécifiquement pour les composants de systèmes

embarqués et même pour les appareils intelligents comme le système **Android**.

4. Commandes de base du terminal Linux

4.1. Le BASH (interpréteur de commandes) et le terminal

Quelle que soit la version de Linux, on trouve une application “terminal” qu’on peut lancer et l’interpréteur de commande avec lequel on interagit est par défaut Bash. (Bourne Again Shell) qui est l’interpréteur de commandes le plus courant sous Linux . Il est également possible de l’activer sous windows 10.



```
mark@linux-desktop: /tmp/tutorial
File Edit View Search Terminal Help
mark@linux-desktop:~$ mkdir /tmp/tutorial
mark@linux-desktop:~$ cd /tmp/tutorial
mark@linux-desktop:/tmp/tutorial$ mkdir dir1 dir2 dir3
mark@linux-desktop:/tmp/tutorial$ mkdir
mkdir: missing operand
Try 'mkdir --help' for more information.
mark@linux-desktop:/tmp/tutorial$ cd /etc ~/Desktop
bash: cd: too many arguments
mark@linux-desktop:/tmp/tutorial$ ls
dir1  dir2  dir3
mark@linux-desktop:/tmp/tutorial$
```

Figure 1: Terminal Linux

Le terminal fonctionne avec le principe **REPL** : **R**ead **E**val **P**rint **L**oop.

1. **Read (Lecture)**. L'utilisateur tape une commande qui est lue par l'interpréteur, généralement une ligne de code ou une commande,
2. **Eval (Évaluation)**. La saisie est ensuite évaluée, ce qui signifie que le code est exécuté ou la commande traitée par l'interpréteur ou le système,
3. **Print (Impression)**. Le résultat de l'évaluation, comme la sortie d'un programme ou la valeur d'une expression, est affiché à l'écran,

4. **Loop (Boucle).** Le processus revient ensuite à l'étape « Lecture », invitant l'utilisateur à saisir une nouvelle saisie.

4.2. Les commandes de base

Toutes ces commandes acceptent de nombreuses options dont on peut consulter la documentation en tapant **man <commande>**.

Par exemple :

- **man ls** permet d'afficher la documentation de la commande **ls**.
- **ls -a** affiche **tous les fichiers**, y compris les fichiers cachés (ceux qui commencent par un point).
- **ls -l** affiche les **détails** des fichiers (droits, propriétaire, taille, date, etc.).

♦ Exemples d'utilisation

Commande	Description	Exemple d'utilisation	Résultat attendu
ls	Lister le contenu du répertoire courant	ls	Affiche les fichiers et dossiers présents
ls -la	Lister tous les fichiers avec détails	ls -la	Montre aussi les fichiers cachés avec leurs permissions
cp	Copier un fichier ou dossier	cp fichier1.txt copie.txt	Crée une copie nommée <i>copie.txt</i>
mv	Déplacer ou renommer un fichier	mv copie.txt Documents/ ou mv fichier.txt nouveau_nom.txt	Déplace le fichier ou change son nom
rm	Supprimer un fichier	rm ancien.txt	Supprime le fichier <i>ancien.txt</i>
cd	Changer de répertoire	cd Documents	Se déplace dans le dossier <i>Documents</i>
cat	Afficher le contenu d'un fichier	cat notes.txt	Affiche le contenu du fichier dans le terminal
echo	Afficher un message ou une variable	echo "Bonjour Linux"	Affiche le texte "Bonjour Linux"
touch	Créer un fichier vide	touch nouveau.txt	Crée un fichier vide nommé <i>nouveau.txt</i>
mkdir	créer des répertoires, aussi appelés dossiers	Mkdir TP01	Crée un dossier vide nommé <i>TP01</i>

Commande	Description	Exemple d'utilisation	Résultat attendu
sudo	Exécuter une commande avec les droits administrateur (superutilisateur)	<code>sudo apt update</code>	Exécute la commande <code>apt update</code> avec les privilèges administrateur

À noter :

- Le terminal distingue **les majuscules et les minuscules** : Documents ≠ documents.
- On peut combiner plusieurs options :
Exemple : `ls -lha` affiche **les fichiers cachés** avec **les détails** et **des tailles lisibles (en Ko, Mo, etc.)**.
- Pour remonter d'un dossier : `cd ..`
- Pour aller directement au dossier personnel : `cd ~`
- Pour aller dans son HOME en utilisant un "chemin absolu", c'est-à-dire un chemin depuis la racine, l'utilisateur bob peut faire : `cd /home/bob/`
- Ainsi, si on veut copier le fichier toto qui se trouve dans le répertoire parent vers le répertoire courant, on tape : `cp ../toto .` Nous utilisons cette fois un chemin "relatif" qui part de la position actuelle.

Exercice pratique

Objectif :

Se familiariser avec les principales commandes de manipulation de fichiers et répertoires sous Linux.

Consignes :

1. Ouvrez un terminal et déplacez-vous dans votre dossier personnel (`cd ~`).
2. Créez un dossier nommé **TP_Linux**.
`mkdir TP_Linux`
3. Entrez dans ce dossier (`cd TP_Linux`).

4. Créez trois fichiers vides : **fichier1.txt**, **fichier2.txt**, **fichier3.txt**.

```
touch fichier1.txt fichier2.txt fichier3.txt
```

5. Listez le contenu du dossier pour vérifier la création (ls).

6. Copiez **fichier1.txt** vers **copie1.txt**.

```
cp fichier1.txt copie1.txt
```

7. Renommez **fichier2.txt** en **ancien.txt**.

```
mv fichier2.txt ancien.txt
```

8. Créez un sous-dossier nommé **Backup**.

```
mkdir Backup
```

9. Déplacez **copie1.txt** dans le dossier **Backup**.

```
mv copie1.txt Backup/
```

10. Affichez le contenu de **fichier3.txt** (vous pouvez d'abord y ajouter un message avec `echo "Bonjour Linux" > fichier3.txt` puis `cat fichier3.txt`).

11. Supprimez le fichier **ancien.txt**.

```
rm ancien.txt
```

✓ Résultat attendu :

À la fin de l'exercice, vous aurez :

TP_Linux/

├── fichier1.txt

├── fichier3.txt

└── Backup/

 └── copie1.txt

4.3. Les répertoires fondamentaux

Dans un système LINUX, on dispose d'une arborescence de fichiers ancrée sur /, la "racine" (root) du système de fichiers.

Voici quelques points d'entrée de cette arborescence :

/

bin ← Commandes de base du système

dev ← Fichiers représentant les dispositifs matériels (devices) du système

etc ← Fichiers de configuration du système

home ← Répertoire d'accueil (HOME) des utilisateurs

lib ← Librairies

mnt ← Points de montage (clés usb etc.)

proc ← État du système et de ses processus

root ← Répertoire de l'administrateur système

run ← Variables d'état du système depuis le boot

sys ← Informations sur le noyau et les périphériques

usr ← Logiciels installés avec le système, base de données etc.

var ← Données fréquemment utilisées et modifiées

4.4. Quelques mots sur BASH

Bash (acronyme de Bourne-Again shell) est un interpréteur en ligne de commande de type script. C'est l'interpréteur de commandes (shell en anglais) Unix du projet GNU.

Usage

Comme tous les interpréteurs en ligne de commande de type script, Bash exécute quatre opérations fondamentales :

- Il fournit une liste de commandes permettant d'opérer sur l'ordinateur (lancement de programmes, copie de fichiers, etc.) ;

- Il permet de regrouper ces commandes dans un fichier unique appelé script ;
- Il vérifie la ligne de commande lors de son exécution ou lors d’une éventuelle procédure de vérification et renvoie un message d’erreur en cas d’erreur de syntaxe ;
- En cas de validation, chaque ligne de commande est interprétée, c’est-à-dire traduite dans un langage compréhensible par le système d’exploitation, qui l’exécute alors.

Les scripts sont de courts programmes généralement faciles à construire. Bash offre un service de gestion de flux, c’est-à-dire qu’il permet que le résultat d’un script (la sortie) soit transmis à un autre script (l’entrée). De cette façon, les scripts peuvent être « chaînés », chacun effectuant une seule tâche bien délimitée.

Les scripts peuvent être exécutés manuellement par l’utilisateur ou automatiquement par le système. Par exemple, dans la distribution GNU/Linux Ubuntu, le répertoire `resume.d` contient un certain nombre de scripts qui s’exécutent automatiquement lors du redémarrage du système, c’est-à-dire après la fin de la mise en veille de celui-ci. Ces scripts servent à relancer les différents programmes interrompus par la mise en veille.

Fonctionnement

Bash est un shell qui peut être utilisé soit en mode interactif, soit en mode batch :

- mode interactif : Bash attend les commandes saisies par un utilisateur puis renvoie le résultat de ces commandes et se place à nouveau en situation d’attente.
- mode batch : Bash interprète un fichier texte contenant les commandes à exécuter.

4.5. Scripts BASH

Une suite de commandes permet à l’administrateur d’automatiser certaines tâches, on parle alors de “scripts”, stocké dans un fichier d’extension “.sh”.

Voici, par exemple un script `efface.sh` qui, au lieu d’effacer le fichier qu’on passe en argument, le déplace dans un dossier “poubelle” à la racine du HOME et l’utilisateur (la première ligne indique l’interpréteur utilisé).

```

#!/bin/bash
# Ce script a un argument : un nom de fichier
# - crée si besoin un répertoire poubelle dans le répertoire HOME
# de l'utilisateur
# - déplace le fichier donné en argument dans ce répertoire
# poubelle
# vérifie d'abord si ce dossier existe dans ~ :
if [ -d ~/poubelle ]
then
    echo "Le répertoire poubelle existe déjà dans votre Home."
else # sinon, on le crée
    mkdir ~/poubelle
    echo "Repertoire poubelle inexistant. Il est crée."
fi
# On déplace dans poubelle le fichier donné en argument
mv $1 ~/poubelle

```

Lancement d'un script et fichiers exécutables

- On peut lancer un script (ou autre) en l'invoquant par son nom s'il est exécutable. On précise parfois que le script est dans le répertoire courant en le précédant d'un ./ devant son nom :
./monfic.sh
- Si le script n'est pas exécutable on peut toujours le lancer en tapant : source monfic.sh
- Ces deux méthodes ne sont pas équivalentes : dans le premier cas, un nouveau shell est créé tandis que dans le second, les commandes du script s'exécutent dans le shell courant.

5. Avantages et inconvénients des environnements libres (cas de GNU/Linux)

5.1 Avantages de Linux et des environnements libres

1. Liberté et gratuité

Linux est distribué sous licence libre (notamment la GPL), ce qui permet à tout utilisateur de **télécharger, installer, copier et modifier** le système sans frais. Cette liberté favorise la diffusion, la personnalisation et l'apprentissage.

2. Sécurité renforcée

Grâce à son architecture multi-utilisateurs et à une gestion stricte des permissions, Linux est **beaucoup moins vulnérable aux virus et aux logiciels malveillants** que les systèmes propriétaires. Les failles de sécurité sont rapidement corrigées par la communauté.

3. Stabilité et fiabilité

Linux est reconnu pour sa **robustesse** : il peut fonctionner pendant des mois, voire des années, sans redémarrage. C'est pourquoi il est largement utilisé sur les **serveurs**, les **supercalculateurs** et dans les **infrastructures critiques**.

4. Performance et légèreté

Linux consomme peu de ressources, ce qui lui permet de tourner efficacement sur des **machines anciennes ou peu puissantes**. Plusieurs distributions (comme Xubuntu ou Lubuntu) sont spécialement conçues pour cela.

5. Personnalisation totale

L'utilisateur peut modifier presque tous les aspects du système : interface graphique, gestionnaire de fenêtres, thème, raccourcis, etc. Cela en fait un système **hautement adaptable** aux besoins personnels ou professionnels.

6. Communauté active et support disponible

La communauté Linux est vaste et dynamique. On trouve facilement de l'aide sur des **forums**, **wikis**, **documentations** et groupes d'utilisateurs. De nombreuses entreprises offrent aussi un support professionnel.

7. Respect de la vie privée et transparence

Étant open source, Linux ne cache pas ce qu'il fait : **le code source est public**, ce qui limite les risques de surveillance ou de collecte de données à l'insu de l'utilisateur.

5.2 Inconvénients de Linux et limites des environnements libres

1. Compatibilité logicielle limitée

Certains logiciels propriétaires (comme Adobe Photoshop, Microsoft Office ou certains jeux récents) ne sont pas disponibles nativement sous Linux. Il faut souvent utiliser des alternatives libres (GIMP, LibreOffice...) ou des outils comme **Wine**.

2. Courbe d'apprentissage plus élevée

Pour les nouveaux utilisateurs, Linux peut sembler complexe, surtout lorsqu'il faut utiliser le **terminal** ou gérer les **permissions**. Cela demande un temps d'adaptation.

3. Compatibilité matérielle variable

Certains matériels (imprimantes, cartes graphiques, périphériques récents) peuvent nécessiter des **pilotes spécifiques** qui ne sont pas toujours disponibles sous Linux.

4. Support technique non centralisé

Bien que la communauté soit active, l'absence d'un service client unique (comme chez Microsoft ou Apple) peut compliquer la résolution de problèmes pour les débutants.

5. Fragmentation des distributions

Il existe des **centaines de distributions Linux**, ce qui peut dérouter les nouveaux venus : Ubuntu, Fedora, Debian, Arch, etc. Chaque distribution a ses outils, sa gestion des paquets, et ses spécificités.

5.3 Bilan

Malgré quelques limites, Linux s'impose aujourd'hui comme un **système puissant, sécurisé et flexible**, idéal pour l'apprentissage, le développement, la recherche et les serveurs. Sa philosophie libre et collaborative en fait un **pilier du monde open source**, et un excellent choix pour ceux qui recherchent **indépendance et maîtrise de leur environnement informatique**.