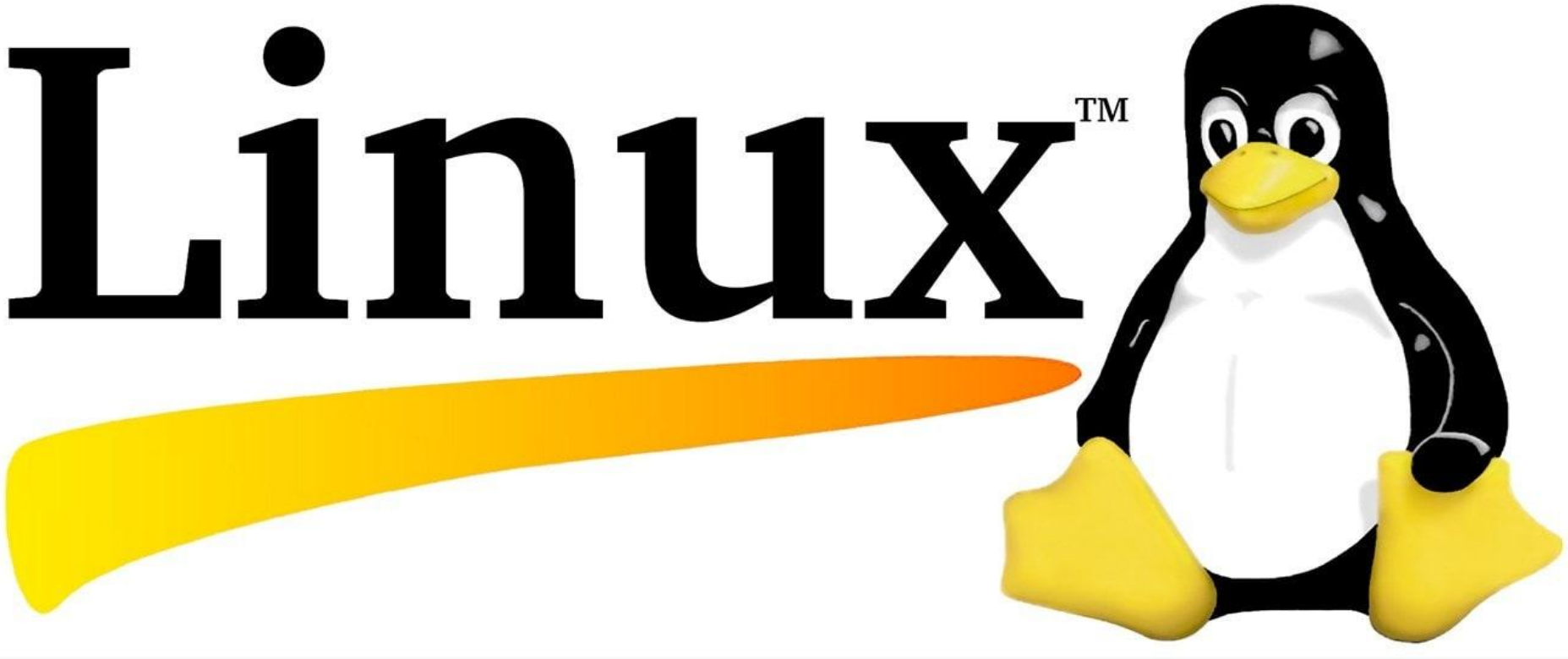
Semaine 1: Introduction à Linux et architecture du système



Semaine 1: Introduction à Linux et architecture du système

Objectifs pédagogiques:

- ➔ Comprendre le contexte historique et les motivations ayant conduit à la création de Linux.
- ➔ Saisir l'évolution du système et l'importance de l'approche communautaire.
- ➔ Découvrir les fondements de la philosophie du logiciel libre et open source.
- ➔ Connaître les principales caractéristiques du système Linux et son architecture (noyau, shell, espaces utilisateur).
- ➔ Appréhender l'arborescence standard d'un système Linux.

Plan détaillé du cours

1. Introduction
2. Contexte de création et évolution du système
3. Philosophie du logiciel libre et open source
4. Principales caractéristiques et architecture de Linux
5. Conclusion et session de questions

📢 **Maîtrisez Linux pour un avenir prometteur !**

Chers étudiants,

Linux n'est pas seulement un système d'exploitation, c'est une **porte d'entrée vers le monde de l'informatique et de l'innovation**. Aujourd'hui, il est **partout** : dans les serveurs web, les systèmes embarqués, les supercalculateurs, les smartphones (Android), et même dans les infrastructures critiques du cloud computing.

Apprendre à utiliser Linux, c'est acquérir des **compétences essentielles** en informatique, en administration système et en cybersécurité. C'est aussi développer une **autonomie** et une **rigueur** dans la gestion des systèmes et des fichiers. Chaque commande que vous maîtrisez est une **brique** qui vous rapproche de l'expertise.

👉 **Pratiquez régulièrement** : Rien ne vaut l'expérience concrète. Testez les commandes, explorez l'arborescence du système et créez des scripts.
👉 **Soyez curieux** : Linux est un monde riche et dynamique. Plus vous explorez, plus vous comprenez son immense potentiel.
👉 **Persévérez** : Comme toute compétence technique, la maîtrise de Linux demande du temps et des efforts. Mais chaque défi relevé renforce votre confiance et votre savoir-faire.

Que vous souhaitiez devenir administrateur système, développeur, ingénieur en cybersécurité ou expert DevOps, **Linux est une base solide** qui vous ouvrira **d'innombrables opportunités**.

🚀 **Alors, prenez ce cours au sérieux, pratiquez sans relâche et faites de Linux un atout majeur dans votre parcours professionnel !** 💡

Votre instructeur, Ariel Nana

Introduction

Pourquoi étudier Linux ?

L'étude de **Linux** est essentielle pour comprendre l'évolution des systèmes d'exploitation modernes, l'impact du **logiciel libre et open source**, ainsi que son rôle fondamental dans l'écosystème technologique d'aujourd'hui. Voici les principales raisons pour lesquelles Linux est important :

1. Importance historique de Linux

A. Un tournant dans l'histoire des systèmes d'exploitation

- **Origines dans UNIX** :
  - Linux s'inspire d'UNIX, un OS développé dans les années 70 qui a posé les bases des systèmes multitâches et multi-utilisateurs modernes.
  - Contrairement à UNIX, qui était souvent propriétaire, Linux a été conçu dès le départ pour être libre et accessible à tous.
- **Création et démocratisation d'un OS libre** :
  - En **1991**, **Linus Torvalds** développe un noyau de système d'exploitation open source pour les PC, permettant aux développeurs et utilisateurs du monde entier de contribuer à son amélioration.
  - Grâce à son **ouverture**, Linux a permis la création d'un écosystème de distributions adaptées aux besoins variés des utilisateurs.

B. L’impact de Linux sur les systèmes modernes

- Linux a influencé les systèmes d’exploitation grand public et professionnels :
  - **MacOS** repose sur un noyau UNIX, montrant l’influence des concepts développés par Linux.
  - **Android**, le système d’exploitation mobile le plus utilisé au monde, est basé sur le noyau Linux.
  - **Windows Subsystem for Linux (WSL)** permet aujourd’hui d’exécuter des applications Linux sur Windows, preuve de la convergence des systèmes.

2. Influence sur l’open source et le mouvement du logiciel libre

A. Un modèle de développement collaboratif

- Contrairement aux systèmes propriétaires (comme Windows ou macOS), Linux est développé par une **communauté mondiale** composée de milliers de contributeurs indépendants, d’entreprises et d’institutions académiques.
- Il repose sur des principes de **transparence, de partage et d’amélioration continue**.

B. Le rôle central des licences open source

- Linux est sous licence **GPL (General Public License)**, qui garantit :
  - L'accès au **code source**.
  - La possibilité de **modifier et redistribuer** le système librement.
  - Une innovation collective sans restriction commerciale.
- Ce modèle a inspiré des milliers d'autres projets open source comme **Apache, MySQL, PostgreSQL, Firefox, Kubernetes et Docker**.

C. Contribution à l'éducation et à la formation en informatique

- Linux est un **excellent outil pédagogique** pour apprendre les bases des systèmes d'exploitation, la programmation, l'administration système et la cybersécurité.
- Il permet aux étudiants et aux professionnels d'expérimenter sans restriction, contrairement aux systèmes propriétaires qui limitent l'accès au code source.

3. Impact technologique et applications modernes

A. Linux domine les infrastructures informatiques modernes

- **Serveurs web et data centers** :
  - Plus de **90 % des serveurs web** fonctionnent sous Linux (via des distributions comme Ubuntu Server, CentOS, Debian, et Red Hat Enterprise Linux).
  - Des entreprises comme **Google, Amazon, Facebook et Microsoft** reposent massivement sur des infrastructures Linux.
- **Supercalculateurs** :
  - **100 % des 500 supercalculateurs les plus puissants** tournent sous Linux en raison de sa **stabilité, flexibilité et optimisation des performances**.
- **Cloud computing et virtualisation** :
  - Linux est au cœur des solutions cloud comme **AWS, Microsoft Azure, Google Cloud et OpenStack**.
  - Les outils de virtualisation comme **Docker et Kubernetes** fonctionnent principalement sur des distributions Linux.

B. Linux et la cybersécurité

- Linux est **le système préféré des experts en cybersécurité** grâce à :
  - Sa transparence (code source auditable).
  - Ses outils avancés (Kali Linux, Parrot OS).
  - Son système de permissions strict.
- **Pourquoi Linux est plus sécurisé ?**
  - Architecture modulaire (meilleure isolation des processus).
  - Gestion avancée des droits d'utilisateur (pas d'accès root par défaut).
  - Mises à jour continues et gestion proactive des failles de sécurité.

C. Linux dans l'IoT et l'embarqué

- Les **dispositifs connectés (IoT)**, comme les voitures autonomes, les routeurs, les caméras de surveillance, utilisent Linux en raison de sa **légèreté et adaptabilité**.
- Exemples :
  - **Android** sur les smartphones et tablettes.
  - **Raspberry Pi** et **Arduino** pour les projets éducatifs et industriels.
  - **Tesla, Mercedes, BMW** utilisent Linux dans leurs systèmes embarqués.

4. Pourquoi apprendre Linux aujourd'hui ?

A. Un atout incontournable pour les carrières en informatique

- **Demande croissante sur le marché du travail** :
  - **Administrateurs systèmes et réseaux** : gestion des infrastructures serveurs.
  - **Développeurs** : intégration de solutions logicielles sous Linux.
  - **Experts en cybersécurité** : analyse et protection des systèmes.
  - **Data scientists et DevOps** : automatisation et gestion des déploiements.
- **Compétences valorisées** :
  - Maîtriser Linux permet d'accéder à **des opportunités professionnelles variées**.
  - De nombreuses certifications existent (LPIC, RHCSA, etc.).

B. Un système adapté aux besoins modernes

- **Flexibilité et personnalisation** :
  - Contrairement aux OS propriétaires, Linux peut être configuré sur mesure pour **s'adapter à n'importe quel usage** (serveurs, bureautique, cloud, embarqué).
- **Coût réduit** :
  - Linux étant **gratuit**, il est accessible à tous, sans frais de licence.
  - Il est largement utilisé dans les entreprises pour réduire les coûts d'infrastructure.
- **Une communauté active et un écosystème en pleine évolution** :
  - La communauté open source continue d'innover avec de nouvelles technologies basées sur Linux.
  - L'avenir des systèmes d'exploitation repose fortement sur l'open source et la collaboration mondiale.

Contexte de création et évolution du système

A. Contexte de création

1. Origines et inspirations

Les systèmes Unix et Minix :

Dans les années 1970 et 1980, **Unix** est l'un des systèmes d'exploitation les plus influents dans le domaine de l'informatique. Développé par **Ken Thompson et Dennis Ritchie** aux Bell Labs en 1969, Unix repose sur plusieurs principes clés :

- **Modularité** : Un système constitué de petits programmes spécialisés qui interagissent.
- **Multitâche et multi-utilisateur** : Plusieurs processus peuvent s'exécuter simultanément.
- **Portabilité** : Facile à adapter à différentes machines grâce à l'utilisation du langage C.

Cependant, Unix est un système **propriétaire**, contrôlé par AT&T, ce qui limite son accès libre et sa modification par d'autres développeurs.

Dans les années 1980, **Andrew S. Tanenbaum** développe **Minix**, un système inspiré d'Unix, destiné à l'enseignement de l'architecture des systèmes d'exploitation. Minix est conçu pour fonctionner sur des **micro-ordinateurs x86** et inclut un **micro-noyau** plus simple et modulaire.
Cependant, Minix est limité en fonctionnalités et **n'est pas réellement libre** (son code source est accessible, mais sous des restrictions qui empêchent des modifications ouvertes).

➡️ **Problème : Il n'existait pas de système libre, puissant et modifiable qui puisse être utilisé sur du matériel grand public.**

2. Les besoins d'un système alternatif

Dans les années 80, les systèmes d'exploitation étaient souvent **fermés et propriétaires** (ex. MS-DOS, Unix commercial). Les chercheurs, étudiants et passionnés avaient peu de contrôle sur ces systèmes.

Les limitations des systèmes existants :
✅ **Unix** : Performant mais coûteux et soumis à des restrictions commerciales.
✅ **Minix** : Gratuit pour l'enseignement, mais avec une licence limitant sa modification.
✅ **MS-DOS** : Pas multitâche et très limité techniquement.

Il fallait donc un **système libre**, performant et **accessible à tous**, qui puisse être développé et amélioré collectivement.

3. Naissance de Linux

Linus Torvalds et le 25 août 1991

Un étudiant finlandais, **Linus Torvalds**, passionné par les systèmes d'exploitation, cherche une alternative à Minix. Il décide alors de **développer son propre noyau** inspiré d'Unix, capable de tourner sur un ordinateur personnel avec un processeur Intel 80386.

Le **25 août 1991**, il publie un message historique sur le forum **comp.os.minix** (Usenet) :

> *« Bonjour à tous ceux qui utilisent Minix - Je suis en train de développer un système d'exploitation (gratuit) pour les compatibles 386 (486). Ce projet est juste un hobby, il ne sera pas aussi grand et professionnel que GNU. »*

Ce message marque la naissance du projet **Linux** !

La première version publique et l'évolution initiale

➡️ En **octobre 1991**, Linus publie la **version 0.01** de Linux, mais sans interface graphique et avec des fonctionnalités très basiques.
➡️ Il met ensuite en ligne la **version 0.12** sous **licence GPL** en 1992, ce qui permet à tout le monde de l'utiliser, modifier et redistribuer.

Rapidement, une **communauté de développeurs** rejoint le projet et améliore le système.
➡️ **GNU/Linux** devient une alternative complète grâce aux outils du projet **GNU** (éditeur de texte, compilateurs, shell, etc.).

B. Évolution du système

1. Croissance communautaire

De projet personnel à projet collaboratif mondial

- **1993** : Plus de 100 développeurs contribuent activement à Linux.
- **1994** : Sortie de **Linux 1.0**, première version stable.
- **1996** : **Tux**, le célèbre pingouin, devient la mascotte officielle de Linux.

Le développement de Linux repose sur une **collaboration ouverte** où chacun peut proposer des améliorations.

➡️ **Modèle de développement distribué** :

- Pas d'entreprise unique derrière Linux.
- Contributions de **milliers de développeurs** et entreprises (IBM, Red Hat, Google, etc.).
- Organisation autour de mainteneurs du noyau (Linus Torvalds gère toujours la branche principale).

2. Les jalons historiques importants

Fusion avec GNU : une étape clé

- En 1983, **Richard Stallman** crée le projet **GNU** pour proposer une alternative libre à Unix.
- En 1992, Linux adopte la licence **GPL** de GNU, ce qui permet son développement ouvert et rapide.
- Linux devient alors un **système d'exploitation complet** :
    - **Noyau Linux** + **Utilitaires GNU** = **GNU/Linux**

3. L'expansion dans le monde de l'entreprise et des serveurs

- Fin des années **1990** : Linux devient une alternative **sérieuse aux systèmes propriétaires** pour les serveurs.
- Apparition des **distributions Linux** :
    - **Debian (1993)**
    - **Red Hat (1994)**
    - **SUSE (1996)**
    - **Ubuntu (2004)**

- **Pourquoi Linux est adopté en entreprise ?**
  - ✅ Gratuité et open source
  - ✅ Sécurité et stabilité
  - ✅ Personnalisation et flexibilité
  - ✅ Performances sur les serveurs

➡️ Aujourd'hui, **Linux est omniprésent dans les datacenters, le cloud et l'embarqué (Android, IoT, supercalculateurs, etc.)**.

4. L'évolution technique de Linux

Améliorations continues du noyau

- Support de **nouvelles architectures matérielles** (ARM, RISC-V, etc.).
- Optimisation du **multitâche et gestion des ressources**.
- **Modules dynamiques** pour améliorer la flexibilité du noyau.

Les innovations apportées par Linux

- **Virtualisation** : Intégration de **KVM (Kernel-based Virtual Machine)** pour les machines virtuelles.
- **Sécurité** : Amélioration avec **SELinux**, AppArmor, et les conteneurs.
- **Performance et efficacité énergétique** : Linux s'adapte aux supercalculateurs et aux appareils basse consommation.
- **Adaptation aux nouvelles technologies** : Linux domine dans le cloud (Kubernetes, Docker), les serveurs et l'IA.

Conclusion de la partie

Résumé des points clés :
✅ Linux est né comme une alternative libre et accessible aux systèmes Unix propriétaires.
✅ Il a évolué grâce à la contribution d'une **communauté mondiale** et au soutien du projet GNU.
✅ Son **architecture robuste et flexible** a favorisé son adoption massive dans l'industrie.
✅ Aujourd'hui, **Linux est omniprésent** : serveurs, supercalculateurs, smartphones (Android), cloud computing, IoT, etc.

Philosophie du logiciel libre et open source

A. Philosophie du logiciel libre

1. Les 4 libertés fondamentales du logiciel libre

La **Free Software Foundation (FSF)**, fondée par **Richard Stallman**, définit un logiciel comme "libre" s'il respecte **quatre libertés essentielles** permettant aux utilisateurs un contrôle total sur leur logiciel :

1. **Liberté d'exécuter le programme comme on le souhaite (Liberté 0)**
   - Un utilisateur peut utiliser le programme **sans restrictions**, quel que soit son usage (personnel, professionnel, éducatif, commercial, etc.).
   - Exemple : Un logiciel libre peut être installé sur n'importe quel ordinateur, qu'il soit personnel ou utilisé dans une grande entreprise.
2. **Liberté d'étudier et de modifier le code source (Liberté 1)**
   - L'accès au **code source** est obligatoire pour permettre aux développeurs et utilisateurs avancés d'étudier le fonctionnement du programme.
   - Toute personne compétente peut modifier le code pour l'adapter à ses besoins.
   - Exemple : Un développeur peut modifier Linux pour améliorer la gestion de la mémoire ou ajouter une nouvelle fonctionnalité.

3. Liberté de redistribuer des copies du logiciel (Liberté 2)

- Un utilisateur peut **partager le programme** avec d'autres, gratuitement ou non.
- Cela favorise la collaboration et permet de diffuser largement des logiciels utiles.
- Exemple : Une école peut copier et distribuer un logiciel libre à tous ses étudiants sans restrictions.

4. Liberté d’améliorer le programme et de publier ses améliorations (Liberté 3)

- Toute amélioration apportée au logiciel peut être partagée avec la communauté.
- Cela permet une **amélioration continue**, car les contributions de chacun bénéficient à tous.
- Exemple : Un développeur améliore la sécurité d’un logiciel libre et publie sa version améliorée pour que d’autres en profitent.

👉 Pourquoi ces libertés sont-elles importantes ?

- Elles permettent aux utilisateurs de **garder le contrôle** sur la technologie qu’ils utilisent.
- Elles favorisent une approche **collaborative et participative** du développement logiciel.
- Elles encouragent l’innovation et l’adaptation des logiciels aux besoins des utilisateurs.

2. Origine et impact du mouvement du logiciel libre

a) Richard Stallman et le projet GNU (1983)

- **Contexte historique :**
  - Dans les années 1970-80, les logiciels deviennent de plus en plus **propriétaires** et fermés.
  - Les utilisateurs n'ont plus accès au code source et ne peuvent plus modifier leurs logiciels.
  - **Richard Stallman**, alors chercheur au MIT, refuse cette évolution et décide de créer une alternative libre.
- **Naissance du projet GNU (1983) :**
  - En 1983, Stallman annonce le projet **GNU** (**GNU's Not Unix**) pour créer un système d'exploitation **entièrement libre**.
  - Objectif : Fournir un système alternatif à UNIX, composé uniquement de logiciels libres.
  - Le développement est long, et en 1991, **Linux** apparaît pour compléter le noyau manquant du projet GNU.

👉 **Impact du projet GNU :**

- GNU fournit les outils et logiciels de base (compilateurs, éditeurs, shells) toujours utilisés aujourd'hui sous Linux.
- Il pose les **fondations juridiques et philosophiques** du logiciel libre.

b) La licence GPL (General Public License)

- La **GPL (General Public License)**, créée en 1989 par Stallman, est la licence la plus utilisée pour les logiciels libres.
- Elle garantit que tout logiciel sous **GPL reste libre**, même si quelqu'un le modifie ou l'améliore.

Principes clés de la GPL :

1. **Le code source doit être disponible.**
   - Un programme sous GPL **ne peut pas être rendu propriétaire**.
2. **Toute modification doit être publiée sous la même licence.**
   - Les améliorations d'un logiciel GPL doivent rester sous GPL.
   - Cela empêche qu'une entreprise privatise un logiciel libre (exemple : un logiciel libre amélioré ne peut pas être vendu sous une licence propriétaire).
3. **La redistribution est autorisée (gratuite ou payante).**
   - On peut vendre un logiciel sous GPL, mais l'acheteur doit aussi recevoir le code source.

👉 **Exemples de logiciels sous GPL :**

- **Linux** : Le noyau Linux est sous licence GPL, ce qui permet à tous de l'améliorer et de l'adapter.
- **GCC** (compilateur C), **GIMP** (logiciel de retouche), **LibreOffice**, etc.

B. Open source : Similarités et différences

1. Définition et principes de l'open source

Le mouvement **open source** partage de nombreuses idées avec le **logiciel libre**, mais il adopte une approche plus **pragmatique** et commerciale.

Caractéristiques du modèle open source :

- **Met l'accent sur la collaboration et l'innovation**, plutôt que sur des principes éthiques.
- **Accepte l'intégration de code open source dans des logiciels propriétaires** (exemple : Android utilise du code open source mais inclut aussi des composants propriétaires).
- **Favorise la transparence et la qualité du code**, car tout le monde peut l'examiner et le tester.

Différences entre logiciel libre et open source :

Logiciel libre	Open source
Basé sur des **principes éthiques et de liberté**	Basé sur la **praticité et l'efficacité**
Respecte obligatoirement les **4 libertés fondamentales**	Autorise l'intégration dans des solutions propriétaires
Défendu par la Free Software Foundation (FSF)	Défendu par l'Open Source Initiative (OSI)
Exemples : GNU/Linux, GIMP, LibreOffice	Exemples : Android, Chromium, Docker

2. Exemples et impacts sur Linux

Le modèle **open source** a joué un rôle clé dans le développement de Linux, en permettant :

- **Une adoption massive** par les entreprises et la communauté.
- **Une amélioration continue**, grâce à la contribution de milliers de développeurs dans le monde.
- **La diversité des distributions** (Debian, Ubuntu, Red Hat, Arch Linux, etc.), adaptées à différents usages.

Exemples de contributions open source sur Linux :

- **Red Hat Enterprise Linux (RHEL)** : Distribution commerciale qui finance des améliorations open source.
- **Android** : Basé sur le noyau Linux, avec des composants open source et propriétaires.
- **Docker et Kubernetes** : Outils de virtualisation et de gestion de conteneurs développés en open source.

3. Discussion interactive

Pour clôturer cette section, **répondez aux questions suivantes** :

- **Connaissiez-vous la différence entre logiciel libre et open source avant cette présentation ?**
- **Selon vous, pourquoi certaines entreprises préfèrent l'open source au logiciel libre ?**
- **Linux aurait-il eu le même succès sans l'open source ?**
- **Quel modèle préférez-vous ? (Libre ou open source) et pourquoi ?**

Principales caractéristiques et architecture de Linux

A. Vue d’ensemble de l’architecture Linux

1. Architecture modulaire

L'architecture de Linux est modulaire, ce qui signifie que le système est construit autour d'un **noyau central** (le kernel), qui communique avec divers composants logiciels et matériels de manière flexible et extensible.

2. Séparation entre le noyau et l’espace utilisateur

L’un des principes fondamentaux de Linux est la séparation entre :

- **Le noyau (Kernel)** : la partie centrale qui interagit directement avec le matériel.
- **L’espace utilisateur (User Space)** : où s’exécutent les applications et les services utilisés par les utilisateurs.

Pourquoi cette séparation ?

- **Sécurité** : Les programmes des utilisateurs ne peuvent pas accéder directement aux ressources matérielles, ce qui évite les erreurs ou les intrusions.
- **Stabilité** : Une panne d’un programme utilisateur ne compromet pas le fonctionnement du noyau.
- **Flexibilité** : L’architecture permet d’exécuter différents environnements et applications sans modifier la structure interne du noyau.

3. Caractéristiques clés de Linux

- **Portabilité** : Fonctionne sur une grande variété de plateformes matérielles (serveurs, PC, smartphones, IoT).
- **Sécurité** : Gestion des permissions avancées, séparation des privilèges et outils de chiffrement.
- **Stabilité** : Conçu pour fonctionner sans interruption pendant de longues périodes (souvent utilisé pour les serveurs).
- **Support matériel étendu** : Compatible avec de nombreuses architectures matérielles grâce à une vaste base de pilotes.

B. Les composants du système Linux

1. Le noyau (Kernel)

Le noyau est le cœur du système d'exploitation et remplit plusieurs fonctions essentielles :

- **Gestion du matériel** : Accès aux périphériques (disque dur, clavier, écran, réseau…).
- **Gestion des processus** : Exécution et planification des tâches.
- **Gestion de la mémoire** : Allocation et protection de la mémoire vive.
- **Gestion des pilotes** : Communication avec les composants matériels.

Structure monolithique vs micro-noyau

- **Noyau monolithique (comme Linux)** :
  - Tous les services essentiels (gestion de la mémoire, processus, fichiers) sont intégrés directement dans le noyau.
  - Avantage : Performance et rapidité.
  - Inconvénient : Une erreur dans le noyau peut impacter l'ensemble du système.
- **Micro-noyau (comme Minix, QNX)** :
  - Sépare les services en modules distincts exécutés en espace utilisateur.
  - Avantage : Plus de stabilité et de sécurité.
  - Inconvénient : Moins performant à cause des interactions fréquentes entre modules.

Linux est un **noyau monolithique modulaire**, ce qui signifie qu'il intègre la gestion des ressources dans le noyau mais permet d'ajouter ou retirer dynamiquement des modules (pilotes, extensions).

2. Le Shell

Le **shell** est l'interface en ligne de commande permettant d'interagir avec le système d'exploitation.

Fonction du shell

- Interpréter et exécuter les commandes utilisateur.
- Automatiser des tâches via des **scripts**.
- Permettre l'administration du système.

Exemples de shells courants

- **Bash** (Bourne Again Shell) : Le plus répandu sur Linux.
- **Zsh** (Z Shell) : Amélioré avec autocomplétion et thèmes.
- **Fish** (Friendly Interactive Shell) : Convivial et interactif.

Exemple de commande :

```
echo "Bonjour, Linux !"
```

3. Les espaces utilisateur (User Space)

L’espace utilisateur est la partie du système où fonctionnent les applications et services.

Applications et services

- **Environnements graphiques** : GNOME, KDE, XFCE.
- **Utilitaires systèmes** : Gestion des fichiers, réseaux, processus.
- **Bibliothèques partagées** : Outils facilitant le développement (ex: glibc).

Interaction avec le noyau

Les applications utilisateur ne communiquent pas directement avec le matériel. Elles passent par :

- **Des appels système (syscalls)** : Interface entre les applications et le noyau.
- **Des bibliothèques** : (ex: libc) pour simplifier les accès au noyau.

C. L’arborescence Linux

1. Présentation de la structure du système de fichiers

Linux suit le **Filesystem Hierarchy Standard (FHS)**, qui définit une organisation logique des fichiers pour faciliter la maintenance et la sécurité.

Pourquoi une norme ?

- Facilite la gestion et la navigation dans le système.
- Uniformise les distributions Linux pour les développeurs et administrateurs.

2. Les répertoires clés

Répertoire	Fonction
`/`	Racine du système, point de départ de toute l’arborescence.
`/bin`	Contient les commandes essentielles (`ls`, `cp`, `mv`, etc.).
`/sbin`	Contient les commandes système réservées aux administrateurs (`fdisk`, `iptables`).
`/usr`	Contient les programmes installés et leurs bibliothèques (`/usr/bin/`, `/usr/lib/`).
`/etc`	Contient les fichiers de configuration système et des services.
`/var`	Stocke des données variables comme les logs et bases de données.
`/home`	Dossiers personnels des utilisateurs.
`/lib` et `/lib64`	Bibliothèques essentielles pour le noyau et les programmes.
`/proc` et `/sys`	Fichiers virtuels affichant l’état du système et du matériel.
`/dev`	Contient les fichiers représentant les périphériques (`/dev/sda1` pour un disque dur).

Merci pour votre attention !!!