ANNÉE PASSERELLE EN INGÉNIERIE

Chapitre 11 — Linux

Introduction à la Téléinformatique

Vincent Audergon · HEIA-FR

Année académique 2026-2027

Plan du chapitre

Objectifs du chapitre

HEIA-FR
HTA-FR

À l'issue de ce chapitre, vous saurez :

- **utiliser** les fonctionnalités de base de Linux : navigation, gestion des fichiers et répertoires, commandes de base ;
- **comprendre** les concepts de base de la gestion des utilisateurs et des permissions ;
- **expliquer** l'arborescence des répertoires et la structure du système de fichiers Linux.

SECTION 01

Qu’est-ce que Linux ?

Un noyau libre et open source

DÉFINITION · LINUX

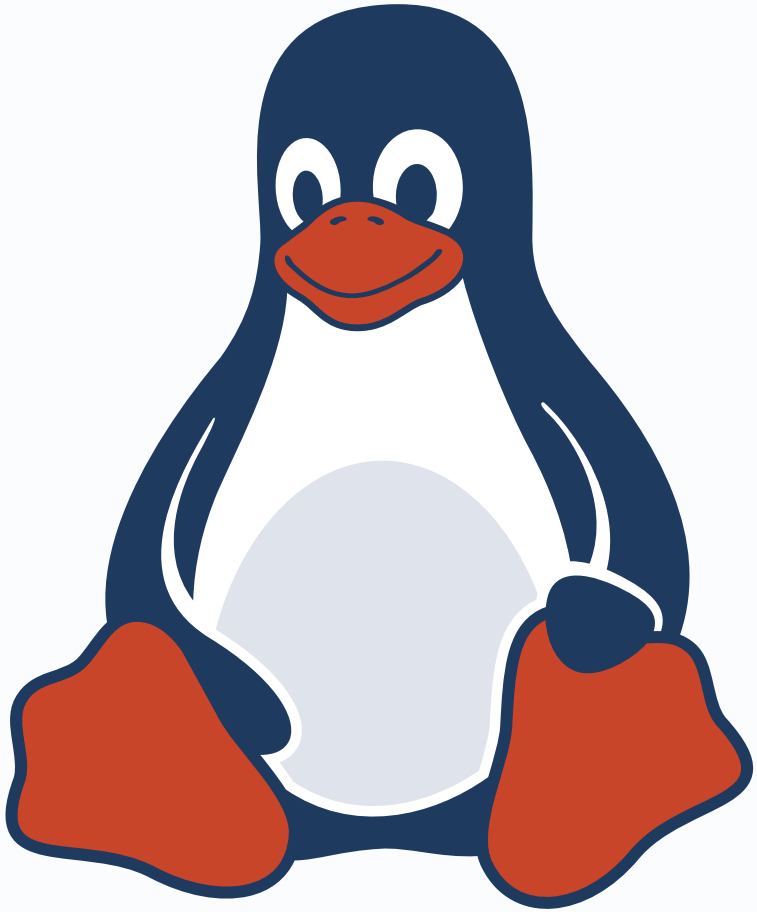
Système d'exploitation **libre** et **open source** basé sur le noyau Linux, créé en **1991** par Linus Torvalds.

- Incontournable, présent partout : serveurs, smartphones (Android), objets connectés, etc.
- Marginal sur le bureau : **3 à 4 %** du marché mondial en 2026.
- Noyau Linux + outils **GNU** = système **« GNU/Linux »**.

REMARQUE

Linux s'utilise surtout en **ligne de commande** (CLI), mais dispose aussi d'environnements graphiques (GNOME, KDE).

Tux



LA MASCOTTE DE LINUX, VOULUE ATTACHANTE ET SYMPATHIQUE

D'Unix à Linux

1969 **Unix**
Ken Thompson et Dennis Ritchie développent Unix aux Bell Labs.

1977 **BSD**
Berkeley publie sa variante d'Unix, ancêtre de macOS.

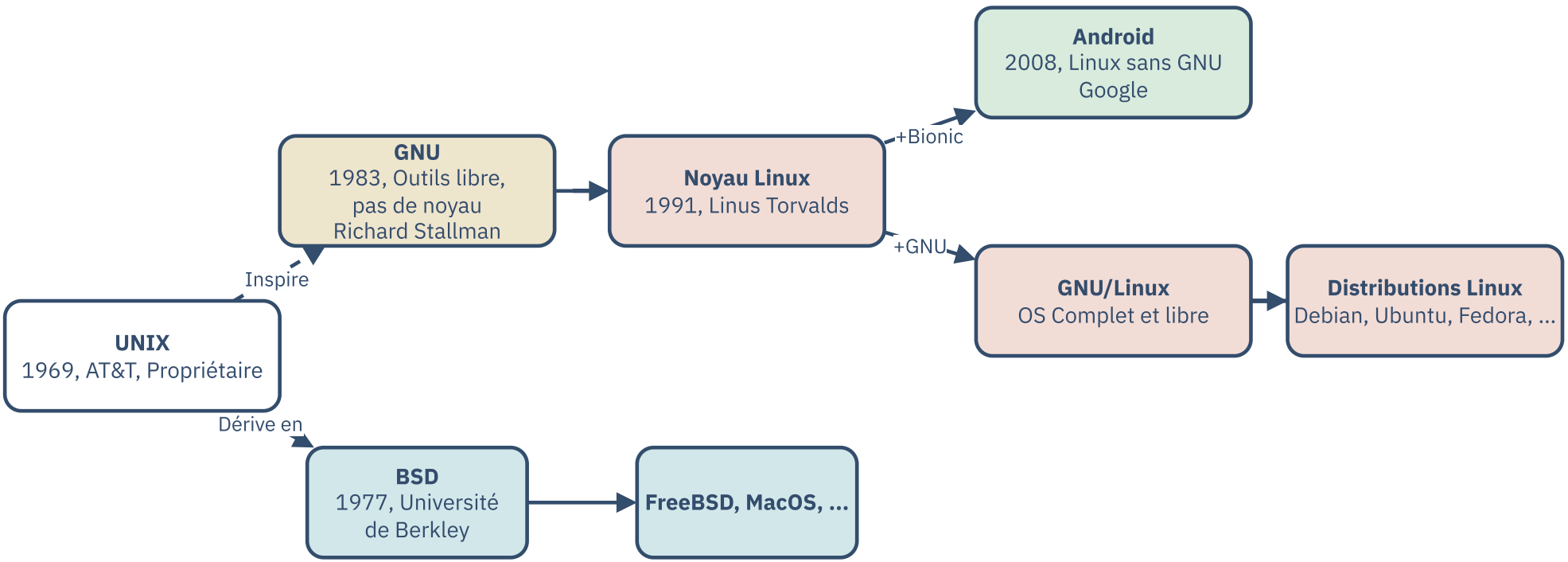
1983 **Projet GNU**
Richard Stallman lance GNU : les outils logiciels (éditeur, compilateur, utilitaires), mais pas de noyau.

1991 **Noyau Linux**
Linus Torvalds publie son noyau, combiné à GNU, un OS libre complet.

1994 **Linux 1.0**
Première version stable, marquant la maturité du noyau.

2008 **Android**
Google publie Android : noyau Linux, sans GNU.

Une famille de systèmes



? QUESTION 1

Quelle est la différence entre le noyau **Linux** et un système **« GNU/Linux »** ?

RÉPONSE

Le noyau **Linux** seul ne suffit pas à créer un système utilisable. Un système **« GNU/Linux »** combine le noyau Linux avec les outils et bibliothèques du projet **GNU** (Bash, GCC, coreutils…). Android, lui, utilise le noyau Linux **sans** GNU.

SECTION
02 —

Distributions et cas d’usage

Le noyau ne suffit pas

HEIA-FR
HTA-FR

REMARQUE

Le noyau Linux et les outils GNU seuls ne forment pas un OS **utilisable** : il faut aussi bibliothèques, applications, gestionnaire de paquets, environnement de bureau… C'est le rôle d'une **distribution** de rassembler tout cela en un système cohérent.

Distribution	Philosophie	Usage typique	Paquets
Ubuntu	Facilité d'utilisation	Bureau, serveurs, cloud	APT
Debian	Stabilité, libre	Serveurs, embarqué	APT
Fedora	Innovation, récent	Bureau, serveurs	DNF
Arch Linux	Contrôle total	Utilisateurs avancés	pacman

Ubuntu

ATTENTION
Ce cours utilise **Ubuntu** comme référence, mais la plupart des concepts et commandes s'appliquent aussi aux autres distributions.

Pourquoi Linux ?

HEIA-FR
HTA-FR

Avantages

- **Gratuit** et open source
- **Sécurité** (multi-utilisateur, permissions strictes)
- **Stabilité** : tourne des mois sans redémarrage
- **Flexibilité** : du serveur à l'embarqué
- **Performances** : léger, rapide, peu gourmand en ressources

Inconvénients

- Courbe d'**apprentissage**
- **Compatibilité** logicielle limitée (jeux, logiciels propriétaires)
- Support technique parfois **communautaire** uniquement

SECTION
03

Système de fichiers et arborescence

Tout est fichier

DÉFINITION · PHILOSOPHIE UNIX

Dans Linux, **tout est considéré comme un fichier**, y compris les périphériques et les informations sur les processus.

- Structure **hiérarchique**, à partir d'un répertoire racine unique : `/`.
- L'utilisateur **root** possède tous les droits ; les autres utilisateurs ont des droits **limités**.
- `sudo` permet d'exécuter une commande avec les privilèges de **root**, sans se connecter en tant que root.

L'arborescence standard

Répertoire	Description
/bin	Programmes exécutables essentiels pour tous les utilisateurs (bin pour « binary »).
/boot	Fichiers nécessaires au démarrage du système.
/dev	Fichiers de périphériques représentant les périphériques matériels du système (dev pour « devices »).
/etc	Fichiers de configuration du système et des applications (etc pour « et cetera », plus tard un « rétro-acronyme » désignera etc comme « Editing Text Config »).
/home	Répertoires personnels des utilisateurs.

L’arborescence standard

Répertoire	Description
/lib	Bibliothèques partagées essentielles pour le fonctionnement des programmes dans /bin et /sbin (lib pour « library »).
/media	Point de montage pour les supports amovibles comme les clés USB. Monter signifie « rendre accessible ».
/mnt	Point de montage temporaire pour monter des systèmes de fichiers manuellement. Un système de fichiers est un ensemble de fichiers et de répertoires organisés sur un support de stockage.
/proc	Système de fichiers virtuel qui fournit des informations sur les processus en cours d’exécution et l’état du noyau.

L'arborescence standard

Répertoire	Description
/root	Répertoire personnel de l'utilisateur root.
/sbin	Programmes exécutables essentiels pour l'administration du système, généralement réservés à l'utilisateur root.
/tmp	Fichiers temporaires créés par les applications et le système. Le contenu de ce répertoire peut être supprimé à tout moment, généralement au redémarrage du système.
/usr	Répertoire d'installation des logiciels (usr pour « user », et plus tard un « rétro-acronyme » désignera usr comme « Unix System Resources »).

? QUESTION 2

Un fichier de configuration système se trouve normalement dans quel répertoire ? Et les fichiers temporaires d'une application ?

> **RÉPONSE**
> Les fichiers de configuration sont dans `/etc`. Les fichiers temporaires sont dans `/tmp`, leur contenu peut disparaître à tout moment, généralement au redémarrage.

SECTION

04 —

Se connecter et naviguer

Se connecter à un système Linux

Deux moyens principaux : directement sur la machine, ou à distance via **SSH** (Secure Shell).

```
user@machine:/some/directory$
```

> **REMARQUE**
> Le symbole `$` indique un utilisateur **non-root**, le symbole `#` indique l'utilisateur **root**.

L'invite est gérée par le **shell**, le plus courant est **Bash** (Bourne Again SHell). D'autres : Zsh, Fish, Ksh.

Terminologie

Quelle différence entre terminal, invite de commande, bash, shell, console ou encore CLI ?

- **Terminal** : interface graphique ou physique pour interagir avec le shell.
- **Shell** : programme qui interprète les commandes (Bash, Zsh…).
- **Invite de commande** : texte affiché par le shell pour indiquer qu'il est prêt à recevoir une commande (affiche typiquement l'utilisateur, la machine et le répertoire courant).
- **Console** : interface texte, souvent utilisée pour l'administration système.
- **CLI** (Command Line Interface) : interface en ligne de commande, par opposition à une interface graphique (GUI).

Qui suis-je ?


```
$ whoami
```

```
darthvader
```

```
$ id
```

```
uid=1001(darthvader) gid=1001(darthvader)
groups=1001(darthvader),100(users)
```

Explication de la commande `id`

```
uid=1001(darthvader) gid=1001(darthvader)
groups=1001(darthvader),100(users)
```

- `uid` : identifiant unique de l'utilisateur
- `gid` : identifiant du groupe principal
- `groups` : liste des groupes auxquels l'utilisateur appartient

Gérer les utilisateurs

- `passwd` : changer un mot de passe
- `su nom` : se connecter en tant qu'un autre utilisateur (su = **substitute user**), nécessite le mot de passe de l'utilisateur cible
- `sudo adduser nom` : ajouter un utilisateur
- `sudo deluser nom` : supprimer un utilisateur

> **ASTUCE**
> Pour découvrir l'utilité d'une commande : `man commande` (manuel complet), `commande --help` (résumé), ou `curl cheat.sh/commande` (fiche en ligne).

Naviguer dans le système de fichiers


```
$ pwd
```

```
/home/darthvader
```

- Chemin **absolu** : commence par `/` (ex. `/home/darthvader/Documents`).
- Chemin **relatif** : relatif au répertoire courant (ex. `Documents`).
- `~` est un alias du répertoire **home** de l'utilisateur courant.

```
$ cd Documents      # changer de répertoire
$ ls -l             # lister avec détails
```

? QUESTION 3

Depuis `/home/darthvader`, quelle est la différence de résultat entre les commandes `cd Documents` et `cd /home/darthvader/Documents` ?

> **RÉPONSE**
> Aucune différence de **résultat** ici : la première utilise un chemin **relatif** (valable seulement depuis le répertoire courant), la seconde un chemin **absolu** (valable depuis n'importe où).

SECTION 05

Gérer fichiers et répertoires

Lire et créer des fichiers


```
$ cat example.txt        # afficher le contenu
$ less example.txt       # lire page par page
$ touch newfile.txt      # créer un fichier vide
$ nano newfile.txt        # éditer (Ctrl+O sauver, Ctrl+X quitter)
```

ATTENTION
`GNU/Linux` est **case-sensitive** : `file.txt` et `File.txt` sont deux fichiers différents.

REMARQUE
`file example.txt` indique le **type** d'un fichier (texte, binaire...).

Créer, supprimer, organiser


```
$ mkdir newdir          # créer un répertoire
$ rmdir newdir          # supprimer un répertoire vide
$ rm -r newdir          # supprimer un répertoire et son contenu
$ rm newfile.txt        # supprimer un fichier
```

ATTENTION

La commande `sudo -rm -rf /` est très connu pour sa dangerosité, à tel point qu'elle est souvent utilisée comme **blague**. Elle supprime **récursivement** et de force tout le système depuis la racine. Les distributions modernes bloquent cette commande par défaut. Mais `--no-preserve-root` permet de forcer quand même ! (à vos risques et périls...)

? QUESTION 4

Quelle commande utiliser pour supprimer un répertoire **non vide** et tout son contenu ?

> **RÉPONSE**
> `rm -r nom_du_repertoire`. L'option `-r` (récursif) supprime le répertoire ainsi que tout son contenu. `rmdir` ne fonctionne, lui, que sur un répertoire **vide**.

SECTION

06 —

Diagnostiquer le réseau

Commandes réseau essentielles

```
$ ping example.com          # tester la connectivité
$ ip addr                   # configuration réseau locale
$ traceroute example.com    # chemin emprunté par les paquets
$ netstat -tuln             # connexions et ports en écoute
$ ip neigh                  # table ARP
```

? QUESTION 5

Selon vous, quel protocole est derrière la commande `traceroute` ?

> **RÉPONSE**
> ICMP (Internet Control Message Protocol), comme `ping`.

Commandes réseau essentielles

EXEMPLE · LECTURE DE IP ADDR

`inet 160.98.22.140/24` → adresse IPv4 de l'interface. `lo` → interface de **loopback** (`127.0.0.1`), non routable sur Internet.

Installer des logiciels (APT)

HEIA-FR
HTA-FR

```
$ sudo apt update                  # rafraîchir la liste des paquets
$ sudo apt install traceroute      # installer un paquet
$ apt search python                # chercher un paquet
$ sudo apt remove traceroute       # désinstaller un paquet
```

REMARQUE

APT est propre à Debian et ses dérivés (Ubuntu, Mint…). D'autres distributions utilisent d'autres gestionnaires de paquets : `dnf` pour Fedora, `pacman` pour Arch Linux…

? QUESTION 6

Dans une sortie `netstat -tuln`, pourquoi les ports TCP affichent-ils le statut « LISTEN » alors que les ports UDP n'en ont pas ?

RÉPONSE
TCP est **orienté connexion** : un serveur doit activement **écouter** les demandes de connexion entrantes. UDP est **sans connexion**. Il n'y a pas de connexion à établir, donc pas d'état « LISTEN ».

SECTION

07 —

Permissions et droits

Propriétaire, groupe, permissions

Chaque fichier a un **propriétaire**, un **groupe**, et des **permissions** pour trois catégories d'utilisateurs.

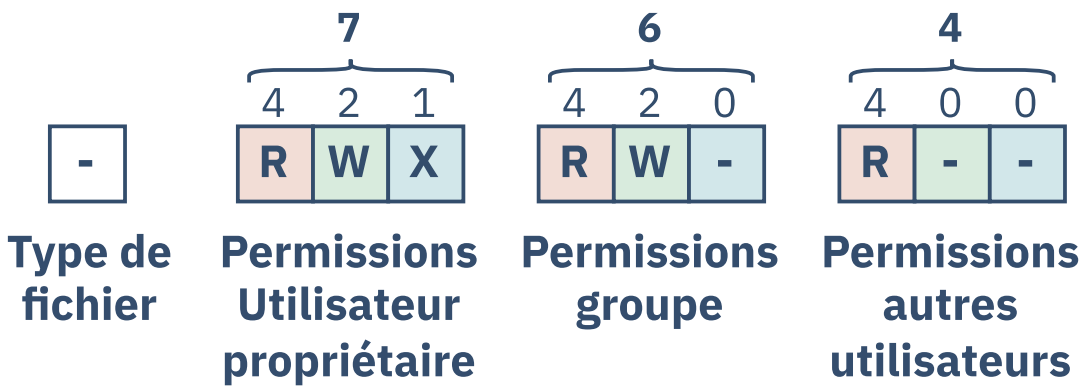
```
$ ls -l
-rw-rw-r-- 1 darthvader darthvader   16 Jul 15 13:09 example.txt
```

- Bloc de 9 caractères = 3 × (propriétaire, groupe, autres).
- `r` lecture, `w` écriture, `x` exécution, `-` absence de permission.

EXEMPLE · EXAMPLE.TXT : RW-RW-R--

Propriétaire et groupe : lecture + écriture. Les autres utilisateurs : lecture seule.

Notation octale



- Lecture = **4**, écriture = **2**, exécution = **1** — un chiffre par catégorie.
- `chmod 764 file.txt` → propriétaire : 7 (rwx), groupe : 6 (rw-), autres : 4 (r–).
- Le caractère `-` initial indique un fichier régulier, `d` un répertoire, `l` un lien symbolique.

Modifier permissions et propriétaire

HEIA-FR
HTA-FR

```
$ chmod u+x example.txt    # notation symbolique
$ chmod g-w example.txt    # retirer w au groupe
$ chmod 640 example.txt    # notation octale
$ sudo chown luke:jedi example.txt    # changer propriétaire et
groupe
```

ASTUCE

Symbolique : `u` (owner), `g` (group), `o` (others), `a` (tous) ; `+` / `-` / `=` pour ajouter, retirer ou fixer une permission.

? QUESTION 7

Que fait la commande `chmod 754 script.sh` ? Détaillez les permissions pour le propriétaire, le groupe et les autres.

> **RÉPONSE**
> **Propriétaire** (7 = 4+2+1) : lecture, écriture, exécution. **Groupe** (5 = 4+1) : lecture, exécution. **Autres** (4) : lecture seulement.

Autres commandes utiles

- `echo` affiche un texte à l'écran.
- `tail` affiche les dernières lignes d'un fichier (ex. `tail -f /var/log/syslog` pour suivre un fichier de log en temps réel).
- `head` affiche les premières lignes d'un fichier.
- `more` et `less` permettent de naviguer dans un fichier page par page.
- `htop` est un moniteur système interactif (processus, CPU, mémoire).
- `tree` affiche l'arborescence des répertoires et fichiers (parfois non installé par défaut).
- `du` (disk usage) affiche l'espace disque utilisé par les fichiers et répertoires.

Enchaîner commandes et redirections

- Enchaînement de commandes avec `|` (pipe) : la sortie d'une commande devient l'entrée de la suivante. Par exemple : `ls -l | less` pour lister les fichiers et les parcourir page par page.
- Redirection de sortie avec `>` (écrase) et `>>` (ajoute) : `echo "Hello" > file.txt` écrit « Hello » dans le fichier, remplaçant son contenu. `echo "World" >> file.txt` ajoute « World » à la fin du fichier.

Synthèse

CE QU'IL FAUT RETENIR

- Linux est un noyau **libre**, combiné aux outils **GNU** et distribué via des **distributions** (Ubuntu, Debian, Fedora...).
- **Tout est fichier**, organisé sous une racine unique `/`.
- Navigation : `pwd`, `cd`, `ls` ; gestion de fichiers : `touch`, `rm`, `mkdir`, `cat`, `nano`. Commandes chaînées avec `|`, redirections avec `>` et `>>`.
- Diagnostic réseau : `ping`, `ip addr`, `traceroute`, `netstat`.
- Permissions : **lecture/écriture/exécution**, **propriétaire/groupe/autres**, modifiables via `chmod` et `chown`.

Des questions ?

Sources

Les sources utilisées pour ce cours (normes, documentation officielle, articles, ouvrages de référence) sont citées et listées intégralement dans la **bibliographie du support de cours écrit**, disponible sur le site du cours :

https://api-teleinfo.isc.heia-fr.ch