

Fiche de révision — Linux : les bases

I. Histoire et filiation Unix

Tout part des **Bell Labs**, laboratoire de recherche financé par les revenus du téléphone d'Alexander Graham Bell. En 1965, Bell Labs, le MIT et General Electric lancent **Multics**, un système multi-utilisateur écrit en assembleur. Le projet est jugé trop lourd, trop lent et trop cher : Bell Labs s'en retire en **1969**.

Ken Thompson porte alors son jeu *Space Travel* sur un **PDP-7** (mini-ordinateur DEC), seule machine libre au labo : le mainframe GE-635 de Multics était facturé au temps machine. En travaillant sur ce PDP-7, Thompson et Dennis Ritchie posent les bases d'un système de fichiers hiérarchique minimaliste. C'est le point de départ d'**Unix**.

En **1972**, Ritchie crée le langage **C** pour réécrire Unix dans un langage portable plutôt que de rester bloqué en assembleur propre à une machine. En **1973**, Unix est réécrit en C, ce qui permet de le porter sur d'autres architectures. Le **code source** est alors livré avec le système : c'est le plan de fabrication, l'ensemble des instructions permettant de reconstruire le logiciel.

***Précision** : le C est un langage **procédural**, pas orienté objet. C'est C++ (Bjarne Stroustrup, 1985) qui ajoute l'orienté objet. Ce que le C apporte à Unix, c'est la **portabilité** : le code n'est plus lié à un jeu d'instructions matériel unique.*

Les embranchements

- **BSD** — l'université de Berkeley développe sa propre variante d'Unix à partir du code source d'AT&T. Il y aura procès. Deux lignées d'Unix coexistent désormais.
- **Minix** — Andrew Tanenbaum, 1987, dans un but pédagogique. Minix introduit la notion clé de **noyau** (**kernel**) : le moteur du système, du code bas niveau au contact direct du matériel, qui gère le processeur, la mémoire, les périphériques et le réseau. Minix tournait sur de très petites machines.
- **Linux** — en **1991**, Linus Torvalds, étudiant à l'université d'Helsinki, développe un noyau sur son 386 SX 25 MHz qui tournait sous Minix, puis l'annonce sur le forum `comp.os.minix` comme un simple projet étudiant.

Richard Stallman et le projet GNU

En 1980, Stallman se heurte à une imprimante Xerox qui bourre en permanence. Il veut modifier le pilote pour signaler les bourrages, mais le code source n'est plus fourni. De cette frustration naît en **1983** le projet **GNU** (**GNU's Not Unix**), dont l'objectif est de reconstruire un système complet de type Unix entièrement libre.

Sa position : la connaissance logicielle doit rester libre, et l'argent se gagne sur le service. La formule exacte est *free as in speech, not free as in beer* : libre comme la liberté d'expression, pas gratuit comme une bière offerte. Le logiciel libre repose sur **quatre libertés** : exécuter, étudier, modifier, redistribuer. Faire payer une licence ou un service reste parfaitement légitime, c'est exactement le modèle défendu.

GNU produit tous les outils système (compilateur, shell, utilitaires) mais son noyau, **GNU Hurd**, n'aboutira jamais en production. Le noyau de Torvalds comble ce trou : c'est la rencontre des deux projets qui donne le système utilisé aujourd'hui, **GNU/Linux**.

Piège d'examen — « **Linux** » désigne strictement le noyau. Une distribution complète, c'est le noyau Linux plus les outils GNU plus un gestionnaire de paquets. D'où l'appellation GNU/Linux.

II. Les distributions

Une **distribution** est un assemblage cohérent : noyau Linux, outils GNU, gestionnaire de paquets, environnement de bureau éventuel, politique de mises à jour. Il en existe plusieurs centaines, presque toutes dérivées de trois souches mères.

Souche mère	Dérivées grand public	Paquets / gestionnaire
Debian	Ubuntu, Linux Mint, Kali, Raspberry Pi OS	.deb / APT

Souche mère	Dérivées grand public	Paquets / gestionnaire
Red Hat	Fedora, CentOS, Rocky Linux, AlmaLinux	.rpm / DNF
Slackware	openSUSE, SUSE Linux Enterprise	.rpm / Zypper

Positionnement des principales distributions

Distribution	Positionnement
Debian	Communautaire, gratuite, très stable, cycle de publication lent. Référence serveur.
Ubuntu	Éditée par Canonical. Gratuite, cycle LTS de 5 ans. Très répandue sur les postes de travail. Le support entreprise (Ubuntu Pro) est le produit payant.
Red Hat Enterprise Linux	Payante. Ce qui s'achète, c'est l'abonnement de support et de maintenance, pas le code. Standard de fait en production.
Fedora	Vitrine amont de Red Hat. Intègre les dernières nouveautés, cycle très court, stabilité potentiellement moindre.
CentOS Stream	Branche amont continue de RHEL depuis 2021. Reçoit les changements avant RHEL, donc moins figée. CentOS Linux 7 est en fin de vie depuis le 30 juin 2024.

Précision : l'analogie « Debian = Windows Home, Ubuntu = Windows Pro » est pédagogique mais fautive dans le détail : Ubuntu est gratuite elle aussi. La bonne opposition est **communautaire gratuit** (Debian, Fedora, Rocky) contre **abonnement de support commercial** (RHEL, SLES, Ubuntu Pro).

Où tourne Linux

- La très large majorité des serveurs web mondiaux tourne sous un système de type Unix, Linux en tête. Internet tourne sous Linux, y compris chez Microsoft (Azure héberge massivement des VM Linux).
- Android** repose sur un noyau **Linux** modifié.
- iOS et macOS** reposent sur **XNU**, un noyau hybride combinant le micro-noyau Mach et des composants BSD. Ce n'est pas un noyau Linux.
- Embarqué, aviation, automobile, routeurs, montres connectées : très majoritairement des noyaux Linux ou BSD.
- Certaines administrations et forces de sécurité utilisent des distributions dérivées internes et fermées (GendBuntu à la Gendarmerie nationale, dérivée d'Ubuntu).

Pourquoi Linux n'est pas partout sur le poste de travail

Un système d'exploitation ne sert qu'à faire tourner des programmes. Les éditeurs développent en priorité pour la plateforme qui concentre les parts de marché poste de travail, c'est-à-dire Windows. La logithèque métier manquante, pas la qualité technique, explique la faible pénétration de Linux sur le desktop. Sur le serveur, où la logithèque est libre, le rapport est inversé.

Appliance — système où le matériel et le logiciel sont packagés ensemble et préconfigurés pour une fonction unique. L'utilisateur n'installe pas d'OS généraliste dessus : il branche et ça fait le travail. Exemple vu en labo : ZeroShell, appliance à noyau Linux orientée routeur, VPN et pare-feu, administrée par interface web uniquement.

III. Arborescence des fichiers

Sous Linux il n'y a **pas de lettres de lecteur** : ni **C:**, ni **D:**. Le point le plus haut de l'arborescence est la racine, notée **/**. Tous les périphériques et partitions sont **montés** quelque part dans cette arborescence unique.

Le séparateur de répertoires est le **slash** **/**, là où Windows utilise l'antislash ****. Cette organisation, normalisée par le **FHS** (**F**ilesystem **H**ierarchy **S**tandard), est commune à toutes les distributions et à tous les dérivés d'Unix.

Répertoire	Contenu
/	Racine de l'arborescence. Tout part de là.
/etc	Fichiers de configuration du système. Aucun binaire exécutable ici, uniquement du texte à éditer.
/bin	Programmes essentiels au démarrage et au dépannage, nécessaires même en mode minimal, avant que /usr ne soit monté.
/sbin	Binaires système réservés à l'administration (ip, fdisk, iptables). Absent du PATH utilisateur par défaut sur certaines distributions.
/usr	Binaires, bibliothèques et ressources système partagées, en lecture seule en usage normal : /usr/bin, /usr/lib, /usr/share, /usr/local.
/home	Répertoires personnels des utilisateurs non-root : /home/pamici. Fichiers personnels et configuration utilisateur.
/var	Données variables : journaux (/var/log), files d'attente, bases de données, sites web (/var/www).
/tmp	Fichiers temporaires, purgés au redémarrage.
/root	Répertoire personnel de l'utilisateur root. À ne pas confondre avec la racine /.

Fichiers de configuration à connaître dans /etc

Fichier	Rôle
/etc/passwd	Liste des comptes utilisateurs. Lisible par tous : les mots de passe chiffrés sont dans /etc/shadow.
/etc/shadow	Empreintes des mots de passe. Lisible par root uniquement.
/etc/group	Liste des groupes et de leurs membres secondaires.
/etc/fstab	Table des montages appliqués au démarrage.
/etc/hosts	Résolution de noms statique, locale à la machine. Consultée avant le DNS selon l'ordre défini dans /etc/nsswitch.conf.
/etc/default/grub	Paramétrage du chargeur de démarrage GRUB.

Étymologie : à l'origine, sur les premiers PDP-11, /usr contenait littéralement les répertoires personnels des utilisateurs, *usr* pour *user*. Le rôle s'est inversé : plus aucun profil utilisateur dedans, tout est passé dans /home. « Unix System Resources » est un rétroacronyme tardif, pas l'étymologie réelle.

IV. Le shell : Bash

Le **shell** est l'interface entre l'utilisateur et le noyau : il lit les commandes, les interprète et demande au noyau de les exécuter. Sous Windows, le shell graphique est le processus `explorer.exe`. Sous Linux, le shell en ligne de commande par défaut est **Bash** : *Bourne Again Shell*.

Le nom rend hommage au **Bourne shell** (`sh`), créé par Stephen Bourne aux Bell Labs en 1977. Bash a été écrit par **Brian Fox** en 1989 pour le projet GNU : un shell libre, compatible `sh`, mais enrichi (historique, complétion, tableaux, arithmétique intégrée). Concrètement, c'est `/bin/bash` : celui qui lit le `~/.bashrc`, exécute les scripts commençant par `#!/bin/bash`, et gère les tubes, les redirections et les variables d'environnement.

Lire l'invite de commande

```
pamici@tuxpa01:~$
```

Élément	Signification
pamici	Nom de l'utilisateur connecté (à gauche du @).
tuxpa01	Nom de la machine (à droite du @).
~	Répertoire courant. Le tilde est un raccourci pour le répertoire personnel, soit /home/pamici.

Élément	Signification
\$	Utilisateur standard. Le symbole devient # lorsqu'on est root.

Règle absolue — Linux est sensible à la casse. `pwd`, `PWD` et `Pwd` sont trois choses différentes. Cela vaut pour les commandes, les fichiers, les dossiers et les variables.

Se documenter

Le manuel intégré est la première source, avant toute recherche web : `man pwd`, `man ls`. Navigation avec les flèches, recherche avec `/motif`, sortie avec `q`.

Arguments courts et arguments longs

Forme	Exemple	Remarque
Argument court	<code>-L</code>	Un seul tiret, une seule lettre. Plusieurs peuvent être cumulés derrière un seul tiret.
Argument long	<code>--logical</code>	Deux tirets, mot complet. Évite que Bash interprète <code>-logical</code> comme le cumul <code>-l -o -g -i -c -a -l</code> .

Premières commandes

Commande	Signification	Rôle
<code>pwd</code>	print working directory	Affiche le chemin absolu du répertoire courant.
<code>ls</code>	list	Liste le contenu d'un répertoire. Équivalent de dir sous Windows.
<code>cd</code>	change directory	Change de répertoire courant.
<code>clear</code>	clear	Efface l'écran du terminal.

Les options de ls à connaître

Option	Effet
<code>-a</code>	Affiche les fichiers et dossiers cachés. Par convention, un élément caché commence par un point.
<code>-l</code>	Format long : type, permissions, propriétaire, groupe, taille, date.
<code>-h</code>	Tailles en unités lisibles par un humain (K, M, G). Ne fonctionne qu'associé à <code>-l</code> .
<code>-R</code>	Parcourt récursivement les sous-répertoires.

Les options se cumulent : `ls -la`, `ls -lah` et `ls -ahl` produisent le même résultat.

V. Chemins et manipulation de fichiers

Chemin absolu et chemin relatif

Type	Point de départ	Reconnaissance
Absolu	Toujours la racine, sans exception. Décrit l'emplacement complet, indépendamment du répertoire courant.	Commence toujours par <code>/</code>
Relatif	Le répertoire courant.	Ne commence jamais par <code>/</code>

Raccourci	Désigne
.	Le répertoire courant.
..	Le répertoire parent.
~	Le répertoire personnel de l'utilisateur courant.
-	Le répertoire précédent (avec cd).

```
cd /home/pamici/Documents      # chemin absolu
cd Documents                  # chemin relatif, depuis /home/pamici
cd ..                         # remonte d'un niveau
cd ~                          # retour au répertoire personnel
```

A. mkdir — créer un répertoire

```
mkdir BACKUP
mkdir -p projet/config/backup # -p crée toute l'arborescence manquante
```

B. touch et nano — créer et éditer un fichier

Commande	Comportement
touch	Crée un fichier vide. N'ouvre jamais d'éditeur, et surtout n'écrase jamais le contenu d'un fichier existant : sur un fichier déjà présent, il met simplement à jour la date d'accès.
nano	Ouvre l'éditeur de texte, et crée le fichier s'il n'existe pas encore, en une seule commande.
cat	Affiche le contenu du fichier dans le terminal, sans l'ouvrir en édition.

Raccourcis nano : **Ctrl+O** enregistrer, **Ctrl+X** quitter, **Ctrl+W** rechercher.

```
touch lettre.txt
nano lettre.txt
cat lettre.txt
```

C. cp — copier

Réflexe d'administrateur : avant toute modification d'un fichier de configuration, on en fait une copie.

```
cp lettreA lettreA.old      # duplique en changeant le nom
cp lettre.txt BACKUP/lettre.txt # copie vers un autre répertoire
cp -r BACKUP/ BACKUP2/      # -r pour copier un répertoire et son contenu
```

D. mv — déplacer et renommer

Une seule commande pour les deux opérations : renommer, c'est déplacer vers le même répertoire sous un autre nom.

```
mv lettreB lettreB.txt      # renommage
mv lettreA.txt BACKUP/      # déplacement
```

E. rm — supprimer

```
rm lettreA.old              # supprime un fichier
rm -r BACKUP/               # -r supprime un répertoire et tout son contenu
```

Danger — il n'y a pas de corbeille en ligne de commande : `rm` supprime définitivement, sans confirmation par défaut. `rm -rf` / détruit le système. Utiliser `rm -i` pour une confirmation à chaque fichier.

VI. Élévation, utilisateurs et permissions

A. sudo — Super User DO

Équivalent fonctionnel de l'UAC et du « Exécuter en tant qu'administrateur » de Windows : `sudo` exécute **une seule commande** avec les privilèges de root, après saisie du mot de passe de l'utilisateur courant. Encore faut-il que cet utilisateur appartienne au groupe `sudo` (Debian et Ubuntu) ou `wheel` (Red Hat et dérivés).

Mise à jour du système

Commande	Effet
<code>sudo apt update</code>	Met à jour l'index local des paquets en interrogeant les dépôts déclarés. N'installe rien : il permet seulement de savoir ce qui est disponible.
<code>sudo apt upgrade</code>	Installe effectivement les mises à jour des paquets déjà présents.
<code>sudo apt install <paquet></code>	Installe un nouveau paquet et ses dépendances.
<code>sudo apt remove <paquet></code>	Désinstalle un paquet en conservant ses fichiers de configuration.

```
sudo apt update && sudo apt upgrade
```

L'ordre est obligatoire : sans `update` préalable, `upgrade` travaille sur un index périmé. Même logique que `ipconfig /release` suivi de `ipconfig /renew` sous Windows.

Précision : `apt` est la commande moderne destinée à l'usage interactif ; `apt-get` reste disponible et est à privilégier **dans les scripts**, car sa sortie et son comportement sont stables entre versions. Les deux fonctionnent, ce ne sont pas des alias.

B. su — Switch User

Commande	Effet
<code>su kiki</code>	Bascule la session vers l'utilisateur kiki. Demande le mot de passe de kiki.
<code>su root</code>	Bascule vers root. Demande le mot de passe de root.
<code>sudo -i</code>	Ouvre un shell root en s'authentifiant avec son propre mot de passe. Préférable à <code>su root</code> , qui suppose de connaître le mot de passe root.
<code>exit</code>	Revient à l'utilisateur précédent.

Attention — root est l'équivalent Linux du compte Administrateur de Windows. Basculer en root, c'est comme activer le compte administrateur local dans la SAM : plus aucun garde-fou, tout devient possible, y compris détruire le système. Sur Debian et Ubuntu, le compte root est d'ailleurs désactivé par défaut, on passe par `sudo`.

Utilisateurs et groupes

Besoin	Commande
Créer un utilisateur	<code>sudo adduser kiki</code>
Créer un groupe	<code>sudo addgroup acdc</code>
Ajouter un membre à un groupe	<code>sudo usermod -aG acdc kiki</code>
Voir les groupes d'un utilisateur	<code>groups kiki</code>
Voir les membres d'un groupe	<code>getent group acdc</code>
Lister les utilisateurs	<code>cat /etc/passwd</code>
Lister les utilisateurs (toutes sources)	<code>getent passwd</code>
Supprimer un utilisateur et son /home	<code>sudo deluser --remove-home kiki</code>

Piège — dans `usermod -aG`, le `-a` signifie *append*. Sans lui, `-G` remplace la totalité des groupes secondaires de l'utilisateur au lieu d'y ajouter le nouveau. La commande est `groups` au pluriel : `group` n'existe pas.

C. Les permissions

Un `ls -l` affiche en tête de ligne une chaîne de dix caractères découpée en blocs de **1 / 3 / 3 / 3**.

```
-rwxrw-r-- 1 pamici acdc 1024 12 août 10:42 lettre.txt

- rwx rw- r--
| | | |
| | | | +-- autres : ni propriétaire, ni membre du groupe
| | | +----- groupe propriétaire
| | +----- propriétaire
+----- type : - fichier, d répertoire, l lien symbolique
```

Position	Bloc	Concerne
1	Type	- pour un fichier, d pour un répertoire (directory), l pour un lien symbolique.
2 à 4	Propriétaire	Droits du propriétaire du fichier.
5 à 7	Groupe	Droits des membres du groupe propriétaire.
8 à 10	Autres	Droits de tous les autres : ni propriétaire, ni membre du groupe.

chmod — modifier les permissions

Chaque droit porte une valeur numérique, et chaque bloc est la somme de ses droits.

Droit	Lettre	Valeur	Sur un fichier	Sur un répertoire
Lecture	r	4	Lire le contenu.	Lister le contenu.
Écriture	w	2	Modifier le contenu.	Créer, renommer, supprimer des entrées.
Exécution	x	1	Exécuter le fichier.	Traverser le répertoire (y entrer).

On raisonne par addition, dans l'ordre **propriétaire, groupe, autres** :

```
chmod 770 lettre.txt
|||
||+-- autres      : 0      = ---
|+--- groupe      : 4+2+1=7 = rwx
+---- propriétaire : 4+2+1=7 = rwx
```

Triplet	Équivaut à	Usage courant
777	rw-rw-rw-	Tout le monde peut tout faire. À proscrire en production.
755	rw-r-xr-x	Répertoires et exécutables : le propriétaire écrit, les autres lisent et traversent.
644	rw-r--r--	Fichiers de données classiques.
600	rw-----	Fichier sensible, lisible du seul propriétaire. Cas des clés privées SSH.
700	rwx-----	Répertoire personnel privé.

chown et chgrp — changer le propriétaire et le groupe

```
sudo chgrp acdc lettre.txt          # change le groupe propriétaire
sudo chown pamici lettre.txt        # change le propriétaire
sudo chown pamici:acdc lettre.txt   # les deux en une commande
sudo chown -R www-data:www-data /var/www/ticket # récursif
```

Annexe — Création d'une VM Linux dans le labo

Paramètre	Valeur retenue
Nom de la maquette	TUXPAxx
Mémoire	4 Go, non dynamique

Paramètre	Valeur retenue
Processeurs	4 cœurs
Commutateur virtuel	Toujours externe, pour permettre la mise à jour du système invité
Disque dur virtuel	Oui

Mémo — commandes à connaître par cœur

Commande	Rôle
<code>pwd</code>	Afficher le répertoire courant.
<code>ls -lah</code>	Lister avec détails, éléments cachés et tailles lisibles.
<code>cd</code>	Changer de répertoire.
<code>mkdir -p</code>	Créer un répertoire et son arborescence.
<code>touch</code>	Créer un fichier vide sans l'écraser.
<code>nano</code>	Éditer un fichier texte.
<code>cat</code>	Afficher le contenu d'un fichier.
<code>cp -r</code>	Copier fichier ou répertoire.
<code>mv</code>	Déplacer ou renommer.
<code>rm -r</code>	Supprimer définitivement.
<code>man</code>	Consulter le manuel d'une commande.
<code>sudo</code>	Exécuter une commande avec les privilèges root.
<code>su</code>	Changer d'utilisateur.
<code>adduser / addgroup</code>	Créer un utilisateur ou un groupe.
<code>usermod -aG</code>	Ajouter un utilisateur à un groupe.
<code>groups / getent</code>	Consulter les appartenances aux groupes.
<code>chmod</code>	Modifier les permissions.
<code>chown / chgrp</code>	Modifier le propriétaire ou le groupe.
<code>apt update / upgrade</code>	Mettre à jour l'index puis les paquets.