

Les systèmes d'exploitation

Rôle, noyau, distributions — logiciel libre et propriétaire

🎯 Objectifs de ce chapitre

- Identifier les fonctions d'un système d'exploitation.
- Distinguer système d'exploitation, noyau et distribution.
- Évoquer les différences entre logiciels libres et logiciels propriétaires.
- Observer ces notions sur un poste Debian, un système d'exploitation libre.

Pourquoi un système d'exploitation ?

Dans les années 1950, un ordinateur ne pouvait exécuter qu'un seul programme à la fois. Pire : chaque programme devait souvent être réécrit pour fonctionner sur une autre machine. Face à ce problème, une idée est née : créer un logiciel intermédiaire, une sorte de pont entre les programmes et le matériel. Ce sont les premiers **systèmes d'exploitation** (en abrégé **OS**, pour *Operating System*).

Vous en utilisez un en permanence sans vous en apercevoir : sur votre téléphone c'est lui qui lance les applications, retrouve vos photos et envoie le son vers le haut-parleur.

Une première approche par cette vidéo : <https://youtu.be/40hUDAtmAuo>

✏️ À toi de jouer

Avec vos propres mots : à quoi sert un système d'exploitation ? Citez deux choses qu'il fait à votre place sur votre téléphone.

Quelques repères historiques

Date	Événement
1945	Les premiers ordinateurs, comme l'ENIAC, n'ont pas de système d'exploitation : on les programme directement.
1956	GM-NAA I/O, l'un des premiers OS, conçu pour l'IBM 704 : il exécute des programmes sur cartes perforées et gère les entrées-sorties.
Début des années 1970	Naissance d' Unix aux laboratoires Bell : multitâche et multi-utilisateur. Il est ensuite réécrit en langage C, créé pour l'occasion.
1981	MS-DOS (Microsoft) est livré avec l'IBM PC et domine l'informatique personnelle.
Années 1980-1990	Arrivée des interfaces graphiques : Mac OS (Apple) et Windows (Microsoft), d'abord au-dessus de MS-DOS puis comme système complet.
1991	Linus Torvalds crée le noyau Linux , diffusé sous licence libre (GNU GPL).
2001	Apple lance Mac OS X (aujourd'hui macOS), un système de type Unix.
2007-2008	Apple adapte son système aux smartphones (iOS), Google diffuse Android , construit sur le noyau Linux.

Les systèmes d'exploitation aujourd'hui

Selon [StatCounter](#), Windows équipe environ **30 %** de l'ensemble des appareils (smartphones compris, où dominant Android et iOS), mais environ **76 %** des ordinateurs de bureau. Tous les autres systèmes courants (Linux, Android, macOS, iOS) sont de type Unix.

Remarque : ces chiffres sont des ordres de grandeur, ils évoluent d'un mois à l'autre.

💡 Pour aller plus loin (culture générale)

Linux est né en 1991 : Linus Torvalds, étudiant à Helsinki, voulait un système de type Unix pour son ordinateur personnel. Il publie son code et des milliers de développeurs, partout dans le monde, y contribuent depuis. Linux est aujourd'hui le système dominant des serveurs et des supercalculateurs. L'histoire de Linux en vidéo : <https://www.youtube.com/watch?v=Za6vGTLp-wg>

I. Les fonctions d'un système d'exploitation

📖 Définition — système d'exploitation

Un **système d'exploitation** (OS) est un logiciel qui sert d'intermédiaire entre le matériel et les applications, donc entre le matériel et l'utilisateur. Il est chargé au démarrage de l'ordinateur et fonctionne tant que la machine est allumée.

Couche	Exemples
Utilisateur	toi, devant l'écran, le clavier et la souris (ou les doigts)
Applications	navigateur Internet, traitement de texte, IDE Python, jeux...
Système d'exploitation	Debian, Windows, macOS, Android... (noyau, bibliothèques, outils)
Matériel	processeur, mémoire, disque ou SSD, clavier, écran, réseau...

Une application ne dialogue jamais directement avec le matériel : elle demande au système d'exploitation de le faire pour elle.

1. Les grandes fonctions

Un système d'exploitation remplit six fonctions principales. La dernière colonne indique ce que vous pourrez utiliser pour l'observer sur un poste Linux : on découvrira ces commandes dans les chapitres suivants.

Fonction	Ce que fait le système	Exemple concret	À observer
Gérer les processus	Lance les programmes et partage le temps du processeur entre eux (multitâche).	Écouter de la musique pendant qu'on écrit un texte.	<code>top</code> , <code>ps</code>
Gérer la mémoire	Donne à chaque programme sa zone de mémoire vive et l'empêche d'utiliser celle des autres.	Un programme qui plante n'arrête pas les autres.	<code>free -h</code>
Gérer les fichiers	Organise les données en fichiers et en répertoires (système de fichiers).	Retrouver un document dans ses dossiers.	<code>pwd</code> , <code>ls</code> , <code>cd</code>
Gérer les périphériques	Communique avec le matériel grâce à des pilotes (drivers).	Brancher une clé USB ou une imprimante.	<code>ls /dev</code>
Gérer les utilisateurs	Identifie chaque utilisateur et contrôle ses droits d'accès.	Un élève ne peut pas modifier les fichiers d'un autre.	<code>id</code> , <code>ls -l</code>

Fonction	Ce que fait le système	Exemple concret	À observer
Fournir une interface	Permet à l'utilisateur de donner des ordres : en mode graphique ou en ligne de commande.	Cliquer sur une icône ou taper une commande.	le terminal

2. Le vocabulaire de base

Terme	Signification
processus	Un programme en cours d'exécution.
multitâche	Capacité à faire fonctionner plusieurs programmes en même temps.
multi-utilisateur	Capacité à gérer plusieurs utilisateurs, chacun avec ses fichiers et ses droits, sans conflit.
pilote (driver)	Petit logiciel qui permet au système de commander un périphérique précis.
interface graphique (GUI)	Fenêtres, icônes et souris. Elle se répand dans les années 1980.
ligne de commande (CLI)	On tape des commandes dans un terminal. Elle reste très utilisée car plus rapide pour certaines tâches.

3. Un OS masque la complexité du matériel

Observe ce programme Python, qui lit un fichier :

```
fichier = open("test.txt")
texte = fichier.readlines()
fichier.close()
```

Schématiquement, ce programme demande au système trois services : ouvrir le fichier, lire son contenu, le fermer. On parle d'**appels système**, souvent faits par l'intermédiaire de bibliothèques de fonctions. Le programmeur n'a pas besoin de savoir si le fichier est sur un SSD, sur une clé USB ou sur un serveur du réseau, ni quel système de fichiers est utilisé (ext4, NTFS, exFAT...) : le même code fonctionne sous Windows, Linux ou macOS.

💡 Pour aller plus loin

Un OS comprend généralement un **noyau**, des **bibliothèques**, une **interface** (graphique ou en ligne de commande) et des **outils système** pour l'administrer. Le noyau gère les processus, la mémoire, les périphériques, le système de fichiers, le réseau et les droits d'accès.

II. Noyau, système d'exploitation, distribution

📖 Définitions — noyau et distribution

Le **noyau** (*kernel*) est le cœur du système d'exploitation : il gère la mémoire, les processus et le matériel, et c'est le seul à dialoguer directement avec ce dernier.

Une **distribution** est un ensemble cohérent, prêt à installer, qui réunit un noyau, des outils système, des bibliothèques, un gestionnaire de paquets (pour installer des logiciels) et souvent un environnement de bureau.

Nom	Nature	Remarque
Linux	noyau	Utilisé partout : serveurs, supercalculateurs, Android...
Debian	distribution	Celle des postes de la salle. Elle installe des logiciels avec la commande <code>apt</code> .
Ubuntu, Fedora, Linux Mint	distributions	D'autres distributions construites autour du même noyau Linux.
Android	système mobile	Noyau Linux, avec des services et applications Google propriétaires.
macOS	système	Propriétaire (Apple), de type Unix.
Windows	système	Propriétaire (Microsoft), qui n'est pas de type Unix.

⚠ Attention au vocabulaire

Rigoureusement, **Linux** désigne seulement le noyau. Les commandes comme `ls` ou `cp` viennent surtout du projet GNU : on parle parfois de GNU/Linux. Dans la vie courante, « je suis sous Linux » veut dire « j'utilise une distribution Linux ».



Essaie, observe

Ouvre un terminal (on apprendra à s'en servir au prochain chapitre) et tape :

```
uname -a
cat /etc/os-release
```

1. Dans le résultat de `uname -a`, repère le nom du noyau et sa version (des nombres séparés par des points).
2. Dans le résultat de `cat /etc/os-release`, repère la ligne `PRETTY_NAME` : quelle distribution et quelle version utilises-tu ?
3. Le mot « GNU/Linux » apparaît dans les deux résultats : relis l'encadré ci-dessus pour comprendre pourquoi.

III. Logiciel libre et logiciel propriétaire

Un logiciel est écrit par des programmeurs sous forme de **code source**. Ce que l'utilisateur a le droit de faire de ce code dépend de la **licence** du logiciel.



Définition — logiciel libre

Un **logiciel libre** garantit à ses utilisateurs quatre libertés :

- l'exécuter pour n'importe quel usage ;
- étudier son fonctionnement et le modifier (son code source est accessible) ;
- en redistribuer des copies ;
- distribuer ses versions modifiées.

Critère	Logiciel libre (ex. Debian, LibreOffice)	Logiciel propriétaire (ex. Windows, macOS)
Code source	Accessible : chacun peut le lire, le modifier et le redistribuer.	Non accessible : seul l'éditeur peut le modifier.
Licence	Par exemple GPL, BSD ou MIT.	Licence propre à l'éditeur, qui limite les usages (copie, modification).
Coût	Souvent gratuit, mais ce n'est pas obligatoire.	Souvent payant, parfois gratuit.
Évolution et support	Communauté de développeurs et entreprises ; mises à jour par un gestionnaire de paquets (apt).	Éditeur unique, qui publie les mises à jour et assure le support.

Remarque : « code ouvert » ne veut pas dire « logiciel sûr ». L'avantage est que n'importe qui peut vérifier le code et corriger les défauts.

⚠ Libre ne veut pas dire gratuit

Un logiciel peut être gratuit sans être libre (beaucoup d'applications gratuites ont un code fermé), et un logiciel libre peut être vendu. Ce qui définit un logiciel libre, ce sont les libertés accordées, pas le prix.

La frontière n'est pas toujours nette :

- **Android** repose sur un noyau libre (Linux), mais beaucoup de ses services et applications sont propriétaires ;
- **macOS** est propriétaire, mais construit sur une base de type Unix en partie libre.

À retenir

📖 L'essentiel

- Un **système d'exploitation** est l'intermédiaire entre le matériel et les applications. Il gère les processus, la mémoire, les fichiers, les périphériques, les utilisateurs, et fournit une interface.
- Le **noyau** est le cœur du système. Une **distribution** réunit un noyau et des logiciels : Linux est un noyau, Debian est une distribution.
- Un **logiciel libre** donne accès au code source et accorde quatre libertés. Libre ne veut pas dire gratuit.

Exercices

Exercice 1 — associe chaque situation à la fonction du système concernée

Fonctions : processus, mémoire, fichiers, périphériques, utilisateurs et droits, interface.

- a. Vous lancez ton navigateur alors que la musique continue de jouer :
- b. Vous branchez une clé USB sur l'ordinateur :
- c. Vous enregistrez un document dans le dossier Documents :
- d. Un camarade essaie d'ouvrir votre répertoire personnel et l'accès lui est refusé :
- e. Vous tapez une commande dans un terminal :

Exercice 2

1. Quelle est la différence entre Linux et Debian ?
2. Un logiciel gratuit est-il forcément libre ? Justifiez avec un exemple.
3. Pourquoi le même programme Python peut-il lire un fichier sous Windows comme sous Linux, sans être modifié ?
4. Android est-il un système entièrement libre ? Nuancez votre réponse.