

# Systèmes d'exploitation : Partie 1

## Objectifs :

- Identifier les fonctions d'un système d'exploitation.
- Utiliser les commandes de base en ligne de commande.
- Gérer les droits et permissions d'accès aux fichiers.
- Les différences entre systèmes d'exploitation libres et propriétaires sont évoquées.

## Introduction :

Qu'est ce qu'un système d'exploitation ?

Pour le comprendre visionner la vidéo suivante :

<https://www.youtube.com/watch?v=4OhUDAtmAUo>

(chaîne Youtube netprof + Histoire des systèmes d'exploitation)

Pour faire l'interface entre le matériel et l'utilisateur, un ensemble de programmes, appelé système d'exploitation ( SE ou OS en anglais : Operating System ), "tournent" donc en permanence sur un ordinateur de façon à faciliter son utilisation.

Les utilisateurs utilisent les applications, qui demandent des ressources au système d'exploitation, qui pilote le matériel informatique, le partageant entre différentes applications et différents utilisateurs.

Le système d'exploitation étant proches de la machine, les plus anciens étaient donc écrits en assembleur; actuellement, ils sont généralement écrits dans un langage à mi-chemin entre le bas et le haut-niveau, comme le langage C.

## Rôle d'un système d'exploitation :

Comment fonctionne de façon générale un système d'exploitation ?

Visionner la vidéo du site lumni.fr :

<https://www.lumni.fr/video/comprendre-ce-que-est-un-systeme-d-exploitation>

Le système d'exploitation contient un ensemble de logiciels système ou noyau qui permettent de :

• **Faire l'interface entre le matériel et l'utilisateur (gestion des périphériques d'entrée/sortie)** mais également les autres applications non système ( jeux, traitement de texte,...).

Exemples :

Communication de l'utilisateur vers le matériel : on appuie sur une touche du clavier et on s'attend à ce que le caractère concerné apparaisse à l'écran ;

Communication du matériel vers l'utilisateur : si un fichier qu'on essaye d'atteindre n'existe pas, on a un message d'erreur.

• **Gérer la mémoire de la machine**, pour que son utilisation devienne "transparente" pour l'utilisateur et les programmes : la notion d'adresse mémoire n'existe plus, on y "accède" alors par la notion de variables dont l'emplacement exact ( RAM, disque dur,...) est complètement inconnue pour l'utilisateur qui n'a donc pas à s'en préoccuper.

Le noyau doit aussi empêcher que différents programmes utilisent la même zone de la mémoire, de façon à "isoler" leurs données.

• **Faire en sorte que tous les programmes puissent s'exécuter de façon équitable** : un processeur ne peut exécuter qu'une seule tâche en même temps; pour donner "l'illusion" que plusieurs programmes (appelés alors processus) peuvent s'exécuter simultanément (ce que l'on appelle le multi-tâches), un programme du système d'exploitation appelé ordonnanceur leur donne accès aux ressources de la machine (processeur(s), mémoires, périphériques...) à tour de rôle et en se succédant très rapidement dans le temps; la durée d'utilisation de ces ressources dépend de la priorité donnée à chaque processus selon son utilisation, son importance,.. mais aucun ne doit "monopoliser" la machine.

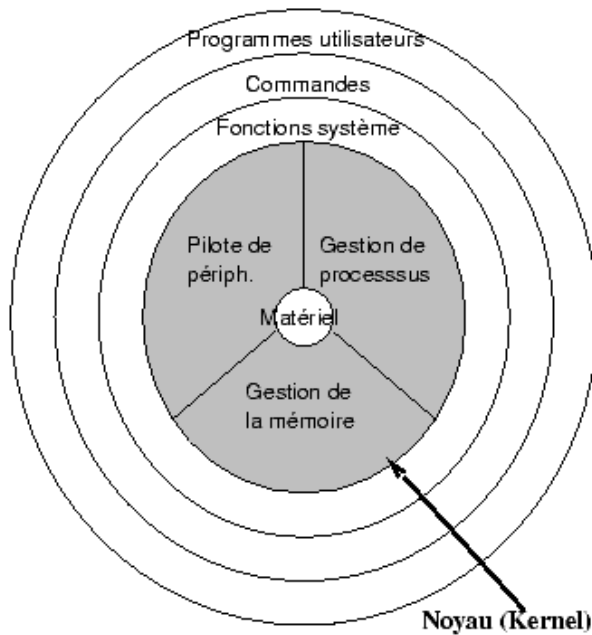
A côté du noyau, un système d'exploitation fournit également d'autres logiciels pouvant réaliser des tâches de plus haut-niveau : programmes utilitaires ou services pour gérer l'affichage, le son, contrôler le système de fichiers,....

• **Isoler le code utilisateur du matériel**, permettant ainsi par exemples :

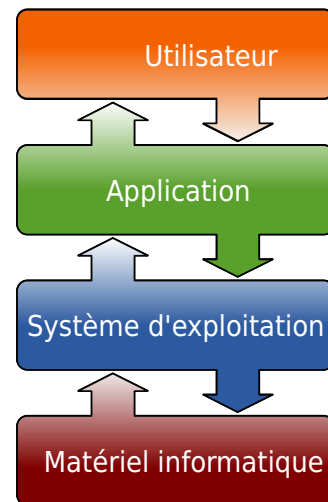
de changer le matériel de façon transparente (ou presque) pour l'utilisateur (il faudra parfois installer des modules ou drivers supplémentaires, mais une fois cela fait, on continue à fonctionner comme précédemment) ;

d'assurer l'intégrité des données et en particulier la gestion des droits et permissions des utilisateurs: par exemple le système assure qu'un utilisateur autre que root (administrateur système) ne peut accéder à des données pour lesquelles il n'a pas les droits d'accès ; ou ne puisse détruire des données...

Le SE est souvent vu comme une succession de "couches" qui s'empilent entre le matériel et l'utilisateur pour faire l'interface entre les deux comme dans les deux représentations ci-dessous :



[http://nsivaugelas.free.fr/premiere/archi\\_s\\_e.php](http://nsivaugelas.free.fr/premiere/archi_s_e.php)



[https://upload.wikimedia.org/wikipedia/commons/e/ed/Operating\\_system\\_placement-fr.svg](https://upload.wikimedia.org/wikipedia/commons/e/ed/Operating_system_placement-fr.svg)

## Les différents systèmes d'exploitation : Libre ou propriétaire

Voici une vidéo qui retrace l'histoire du système d'exploitation **UNIX** :

[https://www.youtube.com/watch?time\\_continue=49&v=bdSWj7Y50VY&feature=emb\\_logo](https://www.youtube.com/watch?time_continue=49&v=bdSWj7Y50VY&feature=emb_logo)

(chaîne Youtube netprof + Histoire du système d'exploitation Unix)

Le système UNIX est un système dit "propriétaire" (certaines personnes disent "privateur"), c'est-à-dire un système non libre. Mais plus généralement, qu'est-ce qu'un logiciel libre ?

D'après Wikipédia : "Un logiciel **libre** est un logiciel dont l'utilisation, l'étude, la modification et la duplication par autrui en vue de sa diffusion sont permises, techniquement et légalement, ceci afin de garantir certaines libertés induites, dont le contrôle du programme par l'utilisateur et la possibilité de partage entre individus". Le système UNIX ne respecte pas ces droits (par exemple le code source d'UNIX n'est pas disponible, l'étude d'UNIX est donc impossible), **UNIX est donc un système "propriétaire"** (le contraire de "libre"). Attention qui dit logiciel libre ne veut pas forcément dire logiciel gratuit (même si c'est souvent le cas), la confusion entre "libre" et "gratuit" vient de l'anglais puisque "free" veut à la fois dire "libre", mais aussi gratuit.

En 1991, un étudiant finlandais, Linus Torvalds, décide de créer un clone libre d'UNIX en ne partant de rien (on dit "from scratch" en anglais) puisque le code source d'UNIX n'est pas public. Ce clone d'UNIX va s'appeler **Linux** (Linus+UNIX). La vidéo ci-dessous raconte l'histoire de Linux en entrant un peu plus dans les détails :

[https://www.youtube.com/watch?v=IquNF\\_DXcF8](https://www.youtube.com/watch?v=IquNF_DXcF8)

(chaîne Youtube Cookie connecté + Linux et Linus - Techtionary #2)

La différence essentielle est que le code d'un logiciel libre (et donc d'un système libre) est public. On peut en général le modifier ou s'en servir pour fabriquer de nouveaux produits.

Les logiciels propriétaires sont en général non ouverts, il est donc plus difficile (voire illégal) de les modifier.

Les logiciels libres sont souvent maintenus par la communauté, mais peuvent aussi l'être par des entreprises qui les utilisent et qui ont intérêt à ce qu'ils restent efficaces et utilisés par d'autres, ce qui assure l'existence de développeurs susceptibles de participer à leur maintien.

Les logiciels propriétaires quant à eux sont essentiellement développés et mis à jour par l'entreprise qui les possède et qui peut décider d'arrêter de les maintenir.

Ce sont deux modèles économiques très différents.

- Systèmes libres : GNU/Linux, GNU/BSD, Android ( pas complètement ),...
- Systèmes propriétaires : Windows, macOS, iOS,...

Les Systèmes d'exploitation libres ont eu pendant longtemps une réputation de systèmes qui ne s'adressent qu'aux "geeks" et autres "cracks" de l'informatique, impossibles à utiliser par "M. Tout-le-monde" du fait de leur complexité; cette réputation est aujourd'hui complètement injustifiée, et ils sont devenus aussi simples à manipuler que les systèmes propriétaires "classiques".

L'utilisation d'un système libre est au programme de la Spécialité NSI.

## IHM et Ligne de commande

### Interface Homme-Machine (IHM)

Une Interface Homme-Machine - IHM est un ensemble de mécanismes, à la fois matériels et logiciels, mis à la disposition des utilisateurs pour leur permettre d'interagir avec un système interactif.

Cette interface est mise en œuvre sur un terminal, ensemble de périphériques d'entrée / sortie, à l'extrémité d'un ordinateur ou d'un réseau. Souvent intégré (clavier, écran tactile, . . . ) mais pas toujours (Terminal bancaire), il peut aussi être virtuel (console d'un système d'exploitation).

Sous Ubuntu, c'est l'application **Terminal** ; sous Windows, **cmd** .

## Bilan :

Comme vous l'avez sûrement compris, définir ce qu'est un système d'exploitation est difficile, mais vous savez maintenant ce qu'il fait, et il est très polyvalent. De manière imagée, il est tout d'abord comme un chef d'orchestre, il s'assure que tous les musiciens et leurs instruments (utilisateurs, processus, logiciels mais également les périphériques physiques : écrans, souris, claviers, imprimantes, bus de liaisons, différentes mémoires...) fonctionnent correctement et de façon harmonieuse, ou au moins synchronisée. Ceci est en particulier le rôle de l'ordonnanceur qui détermine quels processus et quand ceux-ci ont accès aux ressources de calculs du/des processeur(s) et accès aux mémoires. Le système d'exploitation peut aussi être le gardien ou vigile qui s'assure que tout le monde est en règle, à sa place, et a le droit de faire ce qu'il demande. Il est capable de différencier les spectateurs (simples utilisateurs) des musiciens (groupes avec droits particuliers) et surtout du directeur de la salle (l'administrateur : root). Et c'est surtout l'homme de ménage : il range et nettoie après chaque concert (quand vous éteignez votre ordinateur) pour pouvoir reprendre le spectacle rapidement ensuite. Il est aussi ouvreuse, c'est lui qui avant le début du spectacle place les participants et les spectateurs. Pour cela d'ailleurs il est encore aujourd'hui assisté par un autre programme, beaucoup plus 'léger', rapide et nécessitant moins de ressources au démarrage de l'ordinateur. (cf le complément ci-dessous)

Sources :

[https://cache.media.eduscol.education.fr/file/NSI/76/7/RA\\_Lycees\\_G\\_NSI\\_arch\\_systemes\\_appfondis\\_1170767.pdf](https://cache.media.eduscol.education.fr/file/NSI/76/7/RA_Lycees_G_NSI_arch_systemes_appfondis_1170767.pdf)

[https://cache.media.eduscol.education.fr/file/NSI/76/8/RA\\_Lycees\\_G\\_NSI\\_arch\\_systemes\\_utilisateurs\\_1170768.pdf](https://cache.media.eduscol.education.fr/file/NSI/76/8/RA_Lycees_G_NSI_arch_systemes_utilisateurs_1170768.pdf)

[http://nsivaugelas.free.fr/premiere/archi\\_s\\_e.php](http://nsivaugelas.free.fr/premiere/archi_s_e.php)

[https://pixees.fr/informatiquelycee/n\\_site/nsi\\_prem\\_os\\_intro.html](https://pixees.fr/informatiquelycee/n_site/nsi_prem_os_intro.html)

Synthèse de cours de NSI : Mickaël BARRAUD

<https://interstices.info/a-quoi-sert-un-systeme-dexploitation/>

En complément et avant tout système d'exploitation quand on démarre un ordinateur :

## Bios et UEFI

Le BIOS, de l'anglais Basic Input Output System (en français : « système élémentaire d'entrée/sortie ») est un ensemble de fonctions, contenu dans la mémoire morte (ROM) de la carte mère d'un ordinateur, lui permettant d'effectuer des opérations de base, lors de sa mise sous tension. Par exemple l'identification des périphériques d'entrée/sortie connectés et la lecture d'un secteur sur un disque, un CD ou une partie d'une clé USB. Par extension, le terme est souvent utilisé pour décrire l'ensemble du micrologiciel de la carte mère. C'est en quelque sorte le centre de contrôle de la carte mère. Sur les cartes récentes (depuis 2012) il est remplacé par sa version moderne l'UEFI.

Le standard UEFI (de l'anglais Unified Extensible Firmware Interface, signifiant en français : « Interface micrologicielle extensible unifiée ») définit une interface entre le micrologiciel (firmware) et le système d'exploitation (OS) d'un ordinateur. Cette interface succède sur certaines cartes-mères au BIOS. AMD, American Megatrends, Apple, Dell, HP, Intel, IBM, Insyde Software, Microsoft et Phoenix Technologies sont les promoteurs de l'UEFI Forum qui définit les normes de cette technologie

L'UEFI offre quelques avantages sur le BIOS : fonctionnalités réseau en standard, interface graphique de bonne résolution, gestion intégrée d'installations multiples de systèmes d'exploitation et affranchissement de la limite des disques à 2,2 To.

Le BIOS, écrit en assembleur, limitait les modifications et/ou remplacements, gage de sûreté de fonctionnement et de sécurité. L'UEFI est écrit en C, ce qui rend sa maintenance plus souple et reste

acceptable en raison des coûts décroissants de la mémoire. Développé pour assurer l'indépendance entre système d'exploitation et plate-forme matérielle sur laquelle il fonctionne, l'UEFI est disponible sur les plates-formes Itanium (IA-64), x86 (32 et 64 bits) et ARM.

L'UEFI n'est donc pas un simple remplaçant du BIOS. C'est un véritable système d'exploitation.

Sources: [https://fr.wikipedia.org/wiki/BIOS\\_\(informatique\)](https://fr.wikipedia.org/wiki/BIOS_(informatique))  
<https://fr.wikipedia.org/wiki/UEFI>