

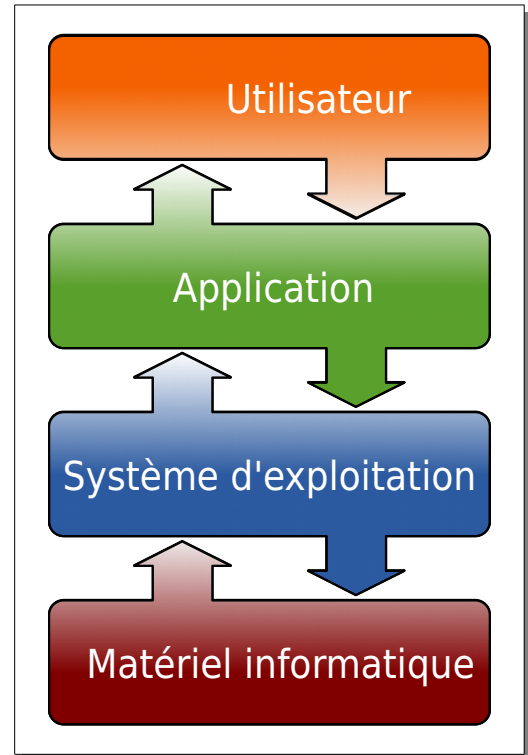
# Système d'exploitation

Programme : Machines  
Architecture matériel et systèmes d'exploitation

## Définition

En informatique, un système d'exploitation (souvent appelé **OS** — de l'anglais **Operating System**) est un ensemble de programmes qui dirige l'utilisation des ressources d'un ordinateur par des logiciels applicatifs. Il reçoit des demandes d'utilisation des ressources de l'ordinateur — ressources de stockage des mémoires (par exemple des accès à la mémoire vive, aux disques durs), ressources de calcul du processeur central, ressources de communication vers des périphériques (pour parfois demander des ressources de calcul au GPU par exemple ou tout autre carte d'extension) ou via le réseau — de la part des logiciels applicatifs. Le système d'exploitation gère les demandes ainsi que les ressources nécessaires, évitant les interférences entre les logiciels.

Le système d'exploitation est un **logiciel**, le deuxième après le firmware et le principal programme exécuté lors de la mise en marche de l'ordinateur, le premier étant le programme d'amorçage (en anglais bootloader). Il offre une suite de services généraux facilitant la création de logiciels applicatifs et sert d'intermédiaire entre ces logiciels et le matériel informatique.



Il existe sur le marché des dizaines de systèmes d'exploitation différents, très souvent livrés avec l'appareil informatique. C'est le cas de Windows, Mac OS, Irix, Symbian OS, GNU/Linux, (pour lequel il existe de nombreuses distributions) ou Android. Les fonctionnalités offertes diffèrent d'un système à l'autre et sont typiquement en rapport avec l'exécution des programmes, l'utilisation de la mémoire centrale ou des périphériques, la manipulation des systèmes de fichiers, la communication, ou la détection et la gestion d'erreurs.

Les deux familles de systèmes d'exploitation les plus populaires sont Unix (dont macOS, GNU/Linux, iOS et Android) et Windows. Cette dernière détient un quasi-monopole sur les ordinateurs personnels avec près de 90 %.

# Histoire des systèmes d'exploitation

**Avant 1950** : Les premiers calculateurs, tels que l'ENIAC (1945), n'ont pas de systèmes d'exploitation. Ils n'exécutent qu'un seul programme à la fois. Ce dernier est saisi en re-cablant physiquement l'ordinateur (en connectant des câbles entre différentes sections faisant des calculs élémentaires).

**1956** : GM-NAA I/O est l'un des premiers systèmes d'exploitation. C'est un programme qui fonctionne sur le calculateurr IBM 704 et dont le rôle est d'exécuter en séquence des programmes utilisateurs stockés sur cartes perforées. Il propose aussi des routines (ou fonctions) pour accéder simplement aux périphériques d'entrée/sortie.

**1967** : MultICS (Multiplexed Information and Computing Service) développé à Bell Labs et au MIT est l'un des premiers systèmes d'exploitation à temps partagé : plusieurs programmes pouvaient s'exécuter « en même temps ». Il a largement influence les systèmes d'exploitation modernes.

**1970-1990** : **Unix**, développé a Bell Labs, est l'un des premiers systèmes d'exploitation multi-taches et multi-utilisateurs. Les premières versions du système sont écrites en assembleurs, puis dans le langage C, nouvellement créé pour cela.



**1978 -1990** : Une variante d'Unix voit le jour et prospère : la Berkeley Software Distribution ou **BSD**.

**1960-1980** : Les ordinateurs du constructeur IBM utilisent diverses variantes du systèmes d'exploitation, différents d'Unix. Ils utilisent en particulièrement à partir des années 1980 le système **MS-DOS** de Microsoft. Ce système est conçu pour fonctionner uniquement sur le processeur x86 d'Intel



**1980-1990** : Les succès de l'architecture x86 d'intel et de l'ordinateur personnel (PC Personal Computer, par opposition aux calculateurs) installent MS-DOS comme le système d'exploitation principal pour les particuliers et de nombreux domaines d'activité.



**1990-présent** : L'éditeur Microsoft développe un système graphique au dessus de MS-DOS, qui deviendra ensuite un système d'exploitation a part entière, le système Windows. Il devient, a la suite de MS-DOS, le système d'exploitation le plus utilisé sur PC.

**1984-2001** : En parallèle de Microsoft, l'entreprise Apple commercialise des ordinateurs personnels basés sur l'architecture Motorola puis Power PC. Les machines sort équipées du système d'exploitation MacOS, un système graphique.

**1991** : Linus Torvalds, alors étudiant a l'Université d'Helsinki, souhaite modifier le système d'exploitation MINIX (une variante d'Unix pour processeurs Intel, développé par Andrew S. Tanenbaum). Bien que les sources de MINIX soient disponibles, la licence logicielle ne permet pas la diffusion des modifications. Linus Torvalds décide de créer son propre système d'exploitation. Il le diffuse sous la licence libre **GNU-GPL**. Le system d'exploitation Linux est né.



**1991-présent** : Le système Linux connaît une adoption rapide, principalement due à sa diffusion sous licence libre. Il est particulièrement répandu comme système d'exploitation pour serveurs et machines de calculs ou de stockage.

**2001-présent** : Apple crée un nouveau système d'exploitation, base sur un système BSD, le système macOS (d'abord appelé OS X).



**2007-présent** : Apple reprend le cœur du système macOS et en développe une version pour les téléphones portables qu'il commercialise, ios.



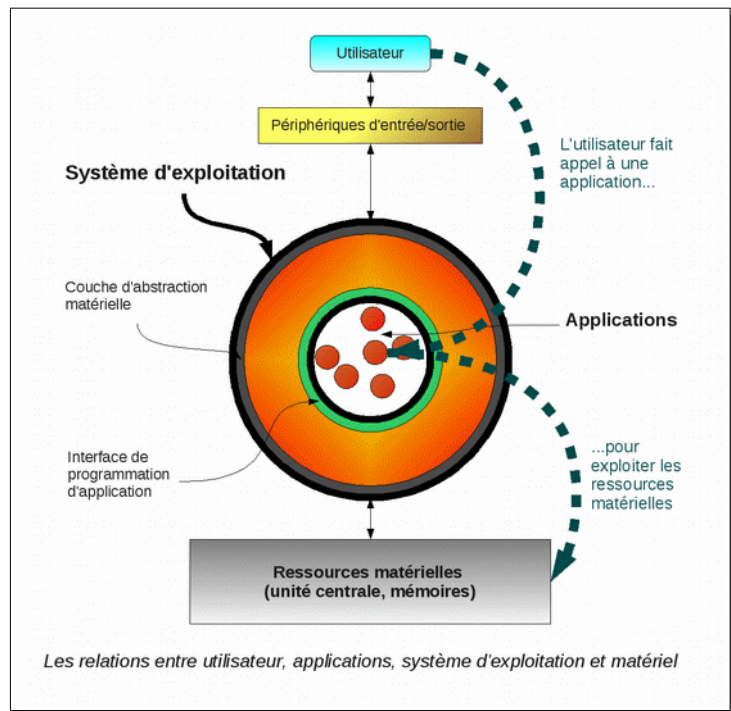
**2008-présent** : Google diffuse le système d'exploitation Android pour téléphones mobiles. Ce système utilise le noyau du système Linux, auquel sont ajoutés des programmes et bibliothèques non libres.



## Terminologie.

**Noyau.** C'est le Cœur du système d'exploitation qui fournit les fonctionnalités basiques aux *programmes*. Il consiste au moins en du code qui est exécuté au démarrage de l'ordinateur et dont le rôle est d'initialiser les périphériques correctement. Le noyau Linux (c'est à dire le programme correspondant au code téléchargeable sur [https : //www.kernel.org](https://www.kernel.org) ) ne contient ni programmes utilisateurs, ni interface (graphique ou texte). Un ordinateur ne possédant que le noyau n'est donc en général pas utilisable en l'état.

**Distribution Logicielle.** C'est un ensemble de logiciels et ressources (fichiers de configuration, icônes, polices de caractères, . . associés à un noyau. En général quand on dit qu'on (( installe Linux » sur un ordinateur, on installe une distribution particulière. Debian, Fedora, Ubuntu, Arch Linux sont des exemples de distributions basées sur le noyau Linux. Il existe plusieurs types de distributions : pour le grand public, adaptées aux machines serveurs, adaptées au matériel embarqué, . . . il existe aussi des distributions pour d'autres noyaux (FreeBSD, GNU/Hurd) .



**Logiciel libre.** Un logiciel est dit libre si son code est couvert par une licence autorisant outre l'utilisation du logiciel, l'accès public au code source, sa modification et la redistribution publique de ces modifications. Parmi les différentes catégories de licence libres, on trouve celles qui autorisent simplement la redistribution des modifications. Ces licences sont dites permissives, par opposition aux licences contraignantes qui obligent l'auteur des modifications à les diffuser sous une licence compatible.

## Le standard POSIX

POSIX est une famille de normes techniques définie depuis 1988 par l'Institute of Electrical and Electronics Engineers (IEEE), et formellement désignée par IEEE 1003. Ces normes ont émergé d'un projet de standardisation des interfaces de programmation des logiciels destinés à fonctionner sur les variantes du système d'exploitation UNIX.

Bien qu'il soit principalement implémenté sur des systèmes de type UNIX, le standard POSIX peut être utilisé par n'importe quel autre système d'exploitation. Par exemple, Microsoft Windows NT 3.x et 4 sont certifiés conformes à POSIX:1990, ce qui est suffisant pour des applications POSIX relativement simples. Cependant, des programmes plus complexes ont besoin d'une compatibilité plus large. Des logiciels supplémentaires tels que Services for UNIX ou Cygwin peuvent apporter à Windows ce niveau de compatibilité.

## Gestion des fichiers

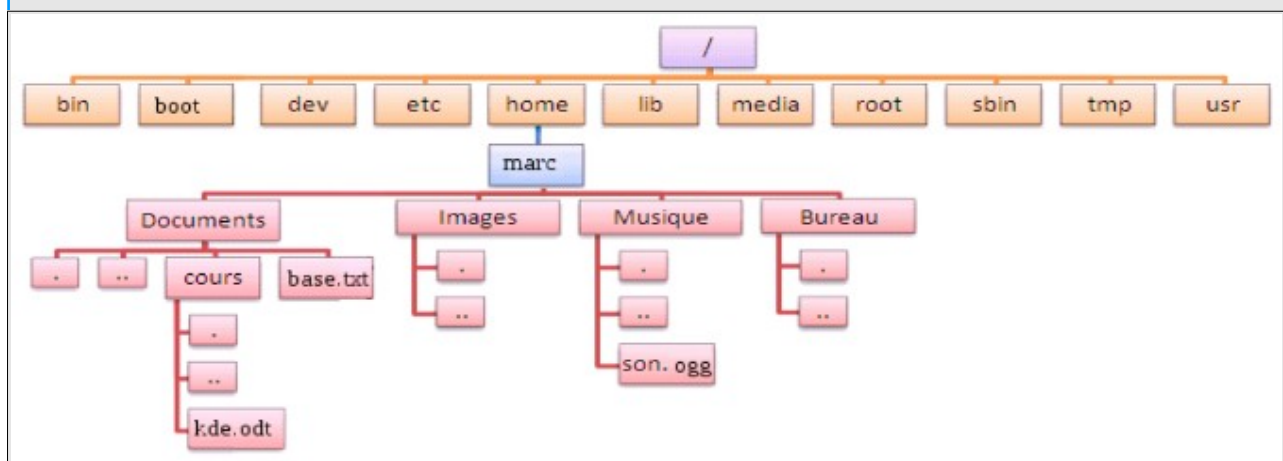
Un système de fichiers est une façon d'organiser et de stocker une arborescence sur un support (disque, disquette, cd ...). Chaque OS a développé sa propre organisation.

On peut faire cohabiter plusieurs systèmes dans des partitions d'un même disque.

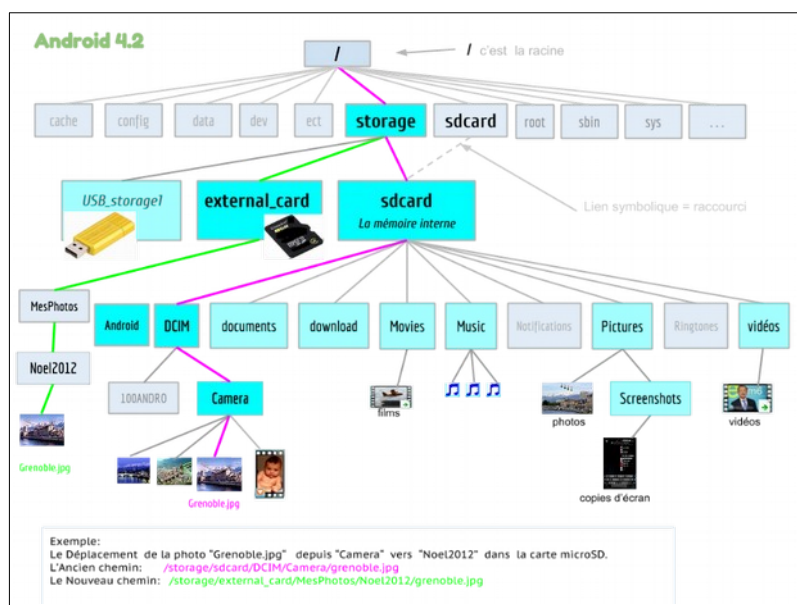
Cette arborescence peut donc être construite (et évoluer) à partir de diverses partitions qui peuvent être situées sur plusieurs disques. Cela réalise une intégration et une abstraction plus poussée que dans le monde Windows où les partitions et lecteurs auxquels sont affectées les lettres A: C: D: ... demeurent des entités séparées.

Sous LINUX, TOUT EST FICHIER  
(dont la racine est nommée / et dont l'administrateur est root)

## Arborescence Linux.



## Arborescence Android



## Bourne shell

Le **Bourne shell** ou `bsh` (ou encore simplement ***sh*** dans de nombreuses versions d'UNIX) est un shell Unix.

Un shell Unix est un ***interpréteur de commandes*** destiné aux systèmes d'exploitation Unix et de type Unix qui permet d'accéder aux fonctionnalités internes du système d'exploitation. Il se présente sous la forme d'une interface en ligne de commande accessible depuis la console ou un terminal. L'utilisateur lance des commandes sous forme d'une entrée texte exécutée ensuite par le shell. Dans les différents systèmes d'exploitation Microsoft Windows, le programme analogue est ***command.com***, ou ***cmd.exe***.

## Utilisateurs et groupes

Toute entité (personne physique ou programme particulier) devant interagir avec un système UNIX est authentifié sur cet ordinateur par un utilisateur ou ***user***. Ceci permet d'identifier un acteur sur un système UNIX. Un utilisateur est reconnu par un nom unique et un numéro unique (la correspondance nom/numéro est stockée dans le fichier `/etc/passwd`). Ce numéro est appelé aussi ***UID***.

Dans les systèmes d'exploitation de type Unix, plusieurs utilisateurs peuvent être catégorisés en groupes. Les permissions sur les fichiers sont organisés en trois classes : ***user***, ***group***, et ***others***.

Un identifiant de groupe, souvent abrégé en ***GID***, est une valeur numérique représentant un groupe spécifique. La commande `id` permet de lister les groupes.

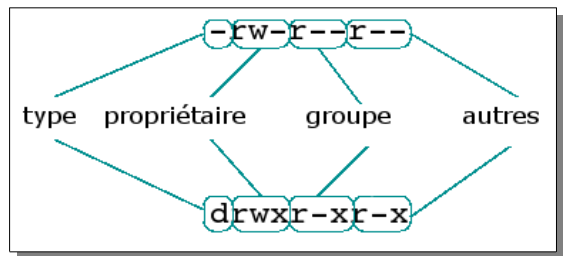
```
steve@steve-Predator-G3-605:~/Bureau/Systeme exploitation$ id steve
uid=1000(steve) gid=1000(steve)
groupes=1000(steve),4(adm),5(tty),7(lp),20(dialout),24(cdrom),27(sudo),30(dip),
,46(plugdev),116(lpadmin),119(saned),126(sambashare)
```

L'utilisation de groupes autorise la délégation de capacités supplémentaires selon un modèle organisé, comme l'accès aux disques, aux imprimantes, et autres périphériques. Cette méthode, parmi d'autres, permet aussi au super-utilisateur de déléguer quelques tâches administratives à des utilisateurs normaux. Cela est similaire aux groupes Administrateurs dans Microsoft Windows.

## Représentation des droits

Cet ensemble de 3 droits sur 3 entités se représente généralement de la façon suivante : on écrit côte à côte les droits `r`, `w` puis `x` respectivement pour le propriétaire (`u`), le groupe (`g`) et les autres utilisateurs (`o`). Les codes `u`, `g` et `o` (`u` comme `user`, `g` comme `group` et `o` comme `others`) sont utilisés par les commandes UNIX qui permettent d'attribuer les droits et l'appartenance des fichiers. Lorsqu'un flag est attribué à une entité, on écrit ce flag (`r`, `w` ou `x`), et lorsqu'il n'est pas attribué, on écrit un `-`.

signifie que le propriétaire peut lire, écrire le fichier, mais que les utilisateurs du groupe attribué au fichier ne peuvent que le lire comme les autres utilisateurs .



Une autre manière de représenter ces droits est sous forme binaire grâce à une clef numérique fondée sur la correspondance entre un nombre décimal et son expression binaire

À l'expression binaire en trois caractères sont associés les 3 types de droits (r w x) ; il suffit donc de déclarer pour chacune des catégories d'utilisateur (user, group, others) un chiffre entre 0 et 7 auquel correspond une séquence de droits d'accès. Par exemple :

777 donne 111 111 111 soit r w x r w x r w x

605 donne 110 000 101 soit r w - - - r - x

644 donne 110 100 100 soit r w - r - - r - -

666 donne 110 110 110 soit r w - r w - r w -

Une astuce permet d'associer rapidement une valeur décimale à la séquence de droits souhaitée. Il suffit d'attribuer les valeurs suivantes pour chaque type de droit :

lecture (**r**) correspond à **4**, écriture (**w**) correspond à **2**, exécution (**x**) correspond à **1**.

Puis on additionne ces valeurs selon qu'on veuille ou non attribuer le droit en correspondant.

Ainsi, rwx « vaut » 7 (4+2+1), r-x « vaut » 5 (4+1) et r- « vaut » 4. Les droits complets (rwxr-xr-) sont donc équivalents à 754. Une manière directe d'attribuer les droits est de les écrire sous cette forme et d'utiliser le code à 3 chiffres résultant avec **chmod**

Utilisation :

Pour voir quels droits sont attribués à un fichier, il suffit de taper la commande

```
1. ls -ll nom_du_fichier
```

```

steve@steve-Predator-G3-605:~/Bureau/Systeme  exploitation$ ls -ll
total 2308
-rw-r--r-- 1 steve steve  71980 sept. 11 17:30  arbolinux.png
-rw-r--r-- 1 steve steve 282228 sept. 11 17:31  'Arborescence simplifiée
Android.png'
-rw-r--r-- 1 steve steve 488653 sept. 11 16:14  bsd.png
-rw-r--r-- 1 steve steve 1411163 sept. 11 17:13  "Les systèmes d'exploitation
POSIX.odt"
-rw-r--r-- 1 steve steve  20733 sept. 11 15:25  Operating_system_placement-
fr.sv
    
```