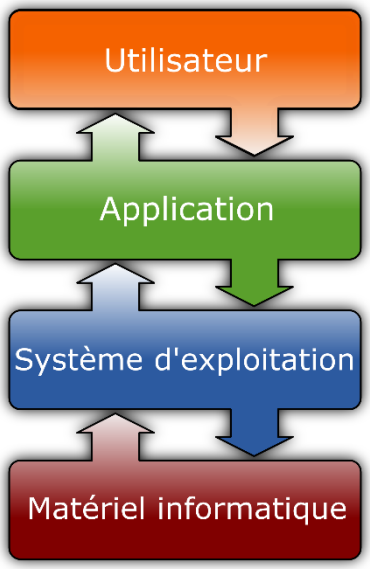
Chapitre 3 : Systèmes d'exploitation

1. Introduction

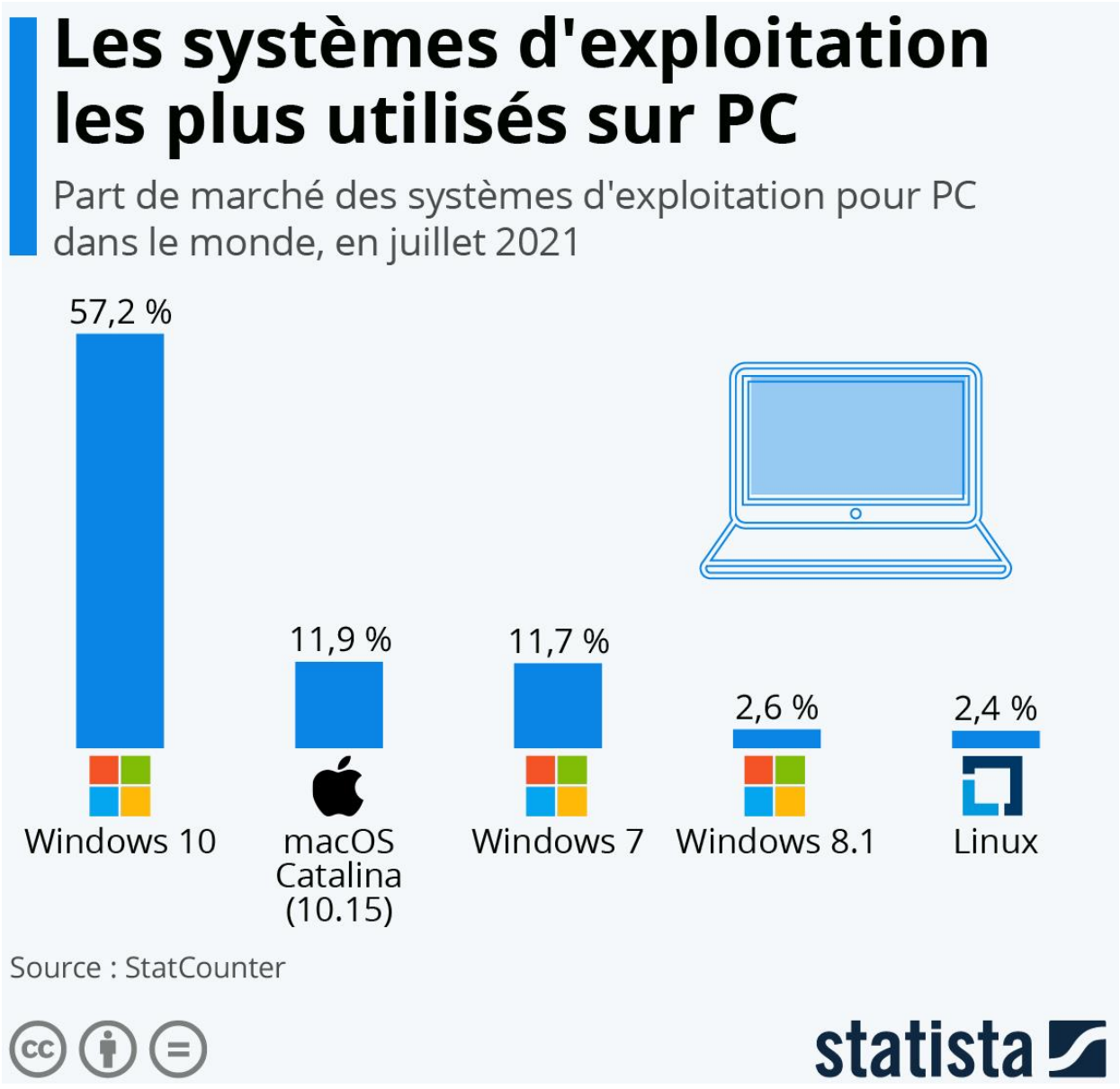
Un système d'exploitation est un programme chargé en mémoire vive dès le démarrage de l'ordinateur, jusqu'à son extinction, dont le rôle est de coordonner l'ensemble des tâches qui sont exécutées par l'ordinateur. Il est situé entre le matériel et les applications (voir ci-contre).



Le système d'exploitation est chargé de :

- Gérer le lancement des diverses applications et donner l'illusion que l'ordinateur est multitâche ;
- Identifier les utilisateurs ;
- Gérer l'organisation du disque dur (ou SSD) et de ses fichiers ;
- Contrôler l'accès aux ressources de stockage (disque dur, SSD, RAM, ROM, …), aux ressources de calcul du (ou des) processeur(s) de l'ordinateur et aux ressources de communication vers les périphériques, les cartes d'extension ou le réseau.

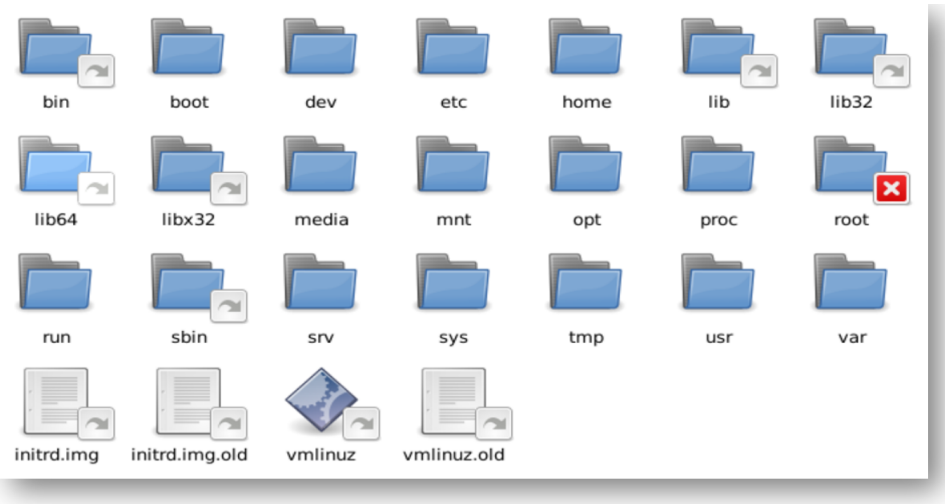
Actuellement, deux grandes familles de systèmes d'exploitation cohabitent : Windows et Unix (avec ses dérivés) :



2. Le système Linux

Linux (plus généralement les systèmes de type Unix) propose un système de fichier en arborescence :

Principaux répertoires d'un système Linux après installation :



« **/** » **:** racine, elle contient les répertoires principaux.

« **/bin** » (binaries) : exécutables essentiels au système, utilisables par tous les utilisateurs (ls, pwd, cp, ...).

« **/boot** » : fichiers permettant à Linux de démarrer (noyau Linux).

« **/home** » : répertoire personnel des utilisateurs.

« **/mnt** » (mount) et « **/media** » : là où les ressources montées apparaissent, de manière permanente (« /media ») ou temporaire (« /mnt »).

« **/root** » : répertoire personnel du super utilisateur (le répertoire de root n'est pas dans /home, car bien souvent le /home est sur une partition à part. En cas d'échec de montage de /home, root à quand même accès à son répertoire personnel).

« **/tmp** » (temporary) : répertoire des fichiers temporaires.

« **/var** » (variable) : contient les données variables (fichiers de log dans /var/log) mais parfois les bases de données (/var/lib/mysql) et les pages de sites web (/var/www/html).

Pour indiquer (ou accéder à) la position d'un fichier (ou d'un répertoire) dans l'arborescence, il existe 2 méthodes :

- Le **chemin d'accès absolu** (slash, « / », en préfixe du chemin) : il indique le « chemin » depuis la racine jusqu'au fichier ou au répertoire ;
- Le **chemin d'accès relatif** (absence de slash initial) : il indique le « chemin » vers un répertoire ou un fichier, en tenant compte du répertoire courant (celui où l'on se trouve).

Remarques :

- Il est tout à fait possible de créer d'autres dossiers dans la racine (par exemple /data), ce n'est pas interdit.
- Les dossiers présentés ci-dessus sont ceux qui sont présents par défaut.
- Le répertoire « **.** » est un répertoire spécial, il désigne le répertoire courant et le répertoire « **..** » désigne le répertoire père du répertoire courant ;
- Dans un chemin relatif, pour remonter d'un niveau dans l'arborescence de fichiers, il faut utiliser deux points « **..** », pour remonter de deux niveaux, on utilise la syntaxe « **../../** », etc.

2.1. Le Bash

Pour interagir avec l'utilisateur, les systèmes d'exploitation disposent d'une **interface graphique** (ex : Windows, MacOs) et d'une **interface en ligne de commande** (console) pour saisir, sous forme textuelle, les commandes que le système d'exploitation doit effectuer.

Quelques commandes de base pour Unix/Linux à connaitre :

Commande	Description - Syntaxe - Exemple
`cd`	Change le répertoire de travail du shell
`cp`	Copie le fichier et les répertoires : `cp source destination`
`ls`	Liste le contenu du répertoire courant
`mkdir`	Crée un répertoire : `mkdir dossier`
`rm`	Supprime un fichier ou un répertoire : `rm fichier  rm -r dossier ou rm -d dossier`
`mv`	Déplace ou renomme un fichier : `mv /home/user/Documents/NSI/nsi.txt /home/user/Musique`
`pwd`	Affiche le répertoire courant
`man`	Retourne le mode d'emploi de la commande s'il existe : `man nom_commande`

2.2. Droits et permissions

Les fichiers et les répertoires possèdent 3 types de droits :

Symbole	Fichier	Dossier
r (read)	Lecture possible du fichier avec un logiciel.	Autorisation d'afficher le contenu du répertoire (la liste des fichiers et des répertoires présents à la racine de ce répertoire)
w (write)	Autorisation de modifier le fichier ou de le vider de son contenu	Autorisation de création, suppression et de changement de nom des fichiers qu'il contient, quels que soient les droits d'accès des fichiers de ce répertoire (même s'ils ne possèdent pas eux-mêmes le droit en écriture)
x (execution)	On peut exécuter le fichier s'il est prévu pour (script, programme)	Autorise l'accès (l'ouverture) au répertoire
–	Le caractère « – » à la place d'un des trois précédents interdit le droit en question	Le caractère « – » à la place d'un des trois précédents interdit le droit en question

Il existe 3 types d'utilisateurs pour un fichier ou un répertoire :

- Le **propriétaire** du fichier (par défaut, la personne qui a créé le fichier), il est symbolisé par la lettre « **u** » (*user*) ;
- Le **groupe** : un fichier est associé à un groupe, tous les utilisateurs appartenant à ce groupe possèdent des droits particuliers sur ce fichier. Il est symbolisé par la lettre « **g** » (*group*) ;
- Les **autres utilisateurs** sont symbolisés par la lettre « **o** » (*others*) ;
- **Tout le monde** est symbolisé par la lettre « **a** » (*all*) : propriétaire + groupe + autres utilisateurs.

→ **L'utilisateur « root » outrepasse tous les droits : il a accès à la lecture et la modification de tous les fichiers**.

Les droits sont représentés par une suite de 9 caractères :

```
-rwxr-xr--
 \ /\ /\ /
  v  v  v
  |  |  droits des autres utilisateurs (o)
  |  |
  |  droits des utilisateurs appartenant au groupe (g)
  |
  droits du propriétaire (u)
```

- Les trois premiers sont les permissions de l'utilisateur ;
- Les trois suivants celles du groupe ;
- Les trois dernières celles des autres utilisateurs.

→ Ici, le propriétaire peut lire, écrire et exécuter le fichier, mais les utilisateurs du groupe attribué au fichier ne peuvent que le lire et l'exécuter. Quant aux autres utilisateurs, ils ne peuvent que lire le fichier.

Exemples :

`chmod u+rw mon_fichier` → Donne au propriétaire les droits en écriture et en lecture au fichier *mon_fichier*.

`chmod -R a+rx mon_dossier` → Donne à tous les utilisateurs les droits en lecture et en exécution à tout ce que contient le dossier *mon_dossier*.
Le « a » est facultatif : `chmod -R +rx mon_dossier` fonctionne tout aussi bien.

Remarques :

❶ Une autre manière de représenter ces droits est d'utiliser une valeur numérique (valeur octale, de base 8) :

- r = **4** si actif ou 0 si inactif
- w = **2** si actif ou 0 si inactif
- x = **1** si actif ou 0 si inactif

Exemples :

`-rwxr-xr--` :
- `rwx` ⇔ **7** (= 4 + 2 + 1)
- `r-x` ⇔ **5** (= 4 + 1)
- `r--` ⇔ **4**

⇒ Les droits complets (rwxr-xr--) sont donc équivalent à **754**.

Droit	Valeur alphanumérique	Valeur octale	Valeur binaire
Aucun droit	`---`	0	000
Exécution seulement	`--x`	1	001
Écriture seulement	`-w-`	2	010
Écriture et exécution	`-wx`	3	011
Lecture seulement	`r--`	4	100
Lecture et exécution	`r-x`	5	101
Lecture et écriture	`rw-`	6	110
Tous les droits (lecture, écriture et exécution)	`rwx`	7	111

Exemples :

`chmod 755 mon_dossier` — Donne au propriétaire tous les droits, aux membres du groupe et aux autres les droits de lecture et d'accès. C'est un droit utilisé traditionnellement sur les répertoires.

`chmod 644 mon_fichier` — Donne au propriétaire les droits de modification et lecture, aux membres du groupe et aux autres uniquement les droits de lecture. C'est un droit utilisé traditionnellement sur les fichiers.

❷ Le premier caractère (« – ») n'est pas un droit, c'est un caractère réservé pour indiquer le type.

Il peut prendre les valeurs suivantes :

- **d** : répertoire
- **l** : lien symbolique
- **c** : périphérique de type caractère
- **b** : périphérique de type bloc
- **p** : fifo (*First in, first out* ou file d'attente)
- **s** : socket
- **–** : fichier classique

Programme :

Contenus	Capacités attendues	Commentaires
Systèmes d'exploitation	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. Les élèves utilisent un système d'exploitation libre. Il ne s'agit pas d'une étude théorique des systèmes d'exploitation.

Bibliographie :

- chmod (Wikipedia.org)

https://pixees.fr/informatiquelycee/prem/c14c.html
https://pixees.fr/informatiquelycee/prem/c14r.html
https://www.linuxtricks.fr/wiki/arborescence-du-systeme-linux

https://www.lyceum.fr/1g/nsi/6-architectures-materielles-et-systemes-dexploitation/3-systemes-dexploitation
https://parc-nsi.github.io/premiere/chapitre9/
https://parc-nsi.github.io/premiere/chapitre9/cours-systeme/systeme-cours-.pdf