NSI – 1ere	COURS Séquence 3B : Operating System	LFV

I. Descriptions et finalités d'un système d'exploitation

1. Introduction.

On peut en tout état de cause dire qu'il existe, pour un utilisateur lambda, deux types d'entités dans un ordinateur (ou de tout autre appareil utilisant un système d'exploitation informatique comme un smartphone, une console de jeux…) :

- Des **fichiers** : ce sont des données stockées de manière pérenne dans l'appareil ; ils apparaissent comme des données plus ou moins **statiques** (photos, films, documents texte mis en forme ou pas…) et peuvent être (mais pas toujours) modifiés grâce à des logiciels.
- Des **processus** : ce sont des logiciels qui effectuent un travail, sans cesse en mouvements ; ils apparaissent comme des données **dynamiques**, très peu contrôlées directement par l'utilisateur.

2. Définition d'un système d'exploitation.

Un système d'exploitation est avant tout un logiciel qui a été programmé, comme tout autre logiciel.
Plusieurs **fonctions** lui incombent :

Il gère les interactions : faire le **lien** entre **l'utilisateur** et le **matériel** :

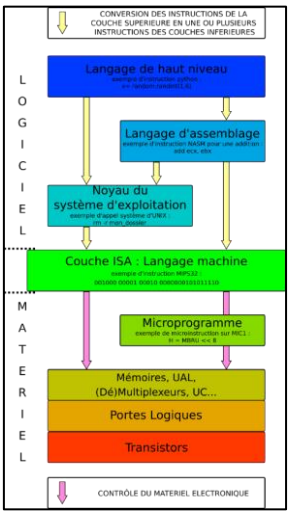
- Communication de l'utilisateur vers le matériel : on appuie sur une touche du clavier et on s'attend à ce que le caractère concerné apparaisse à l'écran ;
- Communication du matériel vers l'utilisateur : si un fichier qu'on essaye d'atteindre n'existe pas, on a un message d'erreur qui nous le communique.

Il gère la persistance : Isoler le code utilisateur du matériel, permettant ainsi par exemple :

- De changer le matériel de façon transparente 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. Le système assure ainsi qu'un utilisateur ne peut pas accéder à des données pour lesquelles il n'a pas les droits d'accès : par exemple les informations qui permettent le démarrage de la machine ou les données d'un autre utilisateur.

Il gère les ressources espace et temps :

- Il attribue les mémoires et les espaces d'adressage (on parle d'*espace* pour les mémoires)
- Faire en sorte que tous les programmes puissent s'exécuter de façon équitable. Un programme qui demande l'utilisation d'une ressource y aura accès en un temps "raisonnable".



3. Où se situe l'OS dans les différentes couches d'un ordinateur :

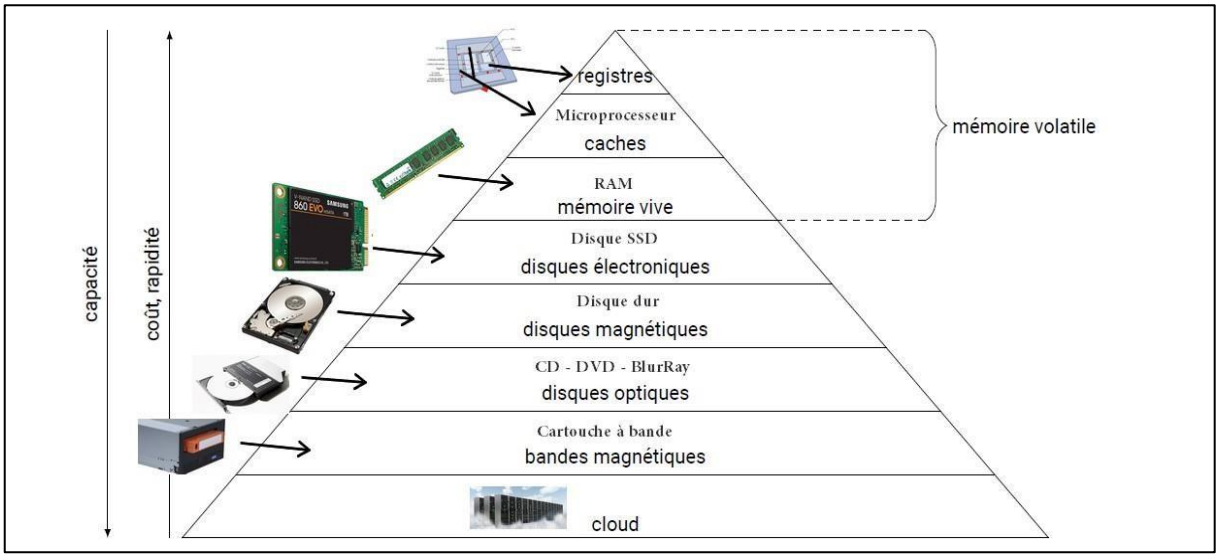
Nous avions vu dans le cours précédent (architecture) le principe des différentes couches d'un ordinateur. Il faut savoir que le système d'exploitation est situé juste au-dessus de la couche ISA qu'il **exploite** pour réussir à effectuer les fonctions évoquées ci-dessus au I.2 (gérer les interactions, la persistance, l'espace et le temps). Fonctionnalités dont n'auront pas à se préoccuper les développeurs de logiciels situés au-dessus du système d'exploitation.

II. Gestion des ressources

1. Les mémoires.

Il existe, dans un ordinateur, différents types de mémoires. Elles différent de part leur rapidité, leur capacité, leur type mais également de leur coût.

Ainsi il est possible de représenter les différentes mémoires d'un ordinateur de la façon suivante.



À l'arrêt de la machine, le contenu de la mémoire volatile disparaît, le contenu des autres mémoires reste.

Détail des mémoires volatiles :

- **« Les registres »,** emplacements mémoire situés au cœur du processeur, très rapides d'accès mais très petits (quelques **dizaines d'octets** en général). **Nous avions par exemple vu – avec l'émulateur IJVM – le registre Program Counter (PC) qui contient l'adresse mémoire de la prochaine instruction à effectuer.** Très souvent modifiés et utilisés, accéder rapidement aux registres est primordial pour l'efficacité du processeur.

- **« Les caches »,** également placé dans le processeur, de l'ordre de quelques Mo aujourd'hui, ils servent à stocker des instructions ou des données temporairement permettant d'accélérer les traitements en limitant les accès à la mémoire vive (accéder au cache est 5 à 10 fois plus rapide qu'accéder à la RAM !)

- **« La mémoire RAM (Random Access Memory) ou mémoire vive »,** mémoire volatile, assez rapide d'accès (taille : de l'ordre du Go aujourd'hui).

Le noyau a intérêt à avoir dans la mémoire la plus rapide d'accès (au plus haut dans le schéma de la figure 1), les données dont il a besoin. Quand le noyau a besoin d'une ressource (un fichier ou un exécutable par exemple), il la copie dans la mémoire volatile. Il travaille sur cette copie et, si besoin, met à jour la mémoire permanente par la suite. C'est ce qui explique que si un ordinateur s'éteint d'un coup et qu'on n'a pas sauvegardé le contenu d'un traitement de texte par exemple, on ne retrouve pas tout ce qui a été écrit au moment du redémarrage.
Malheureusement, plus la mémoire est rapide, plus elle est chère et donc plus sa capacité est réduite. Il est donc impossible de mettre les contenus de tous les fichiers par exemple dans la mémoire volatile ; une partie du travail du système d'exploitation est de gérer le contenu de la mémoire volatile, mais aussi de faire en sorte que les applications n'aient pas à se préoccuper de savoir où se trouvent les données qu'elles traitent (rôle des couches mentionné au I.3).

Remarque : *il existe aussi un type de mémoire particulière non mentionnée ci-dessus qui stocke les informations nécessaires au démarrage d'un ordinateur (avant que la RAM ne soit chargée) ou les micro-programmes du processeur et ne peut pas être effacée ou modifiée.* ***Il s'agit de la mémoire ROM*** *(Read Only Memory) qui est soudée à la carte-mère de l'ordinateur et est encodée une seule fois : lors de la fabrication de la machine !*

2. Les processus.

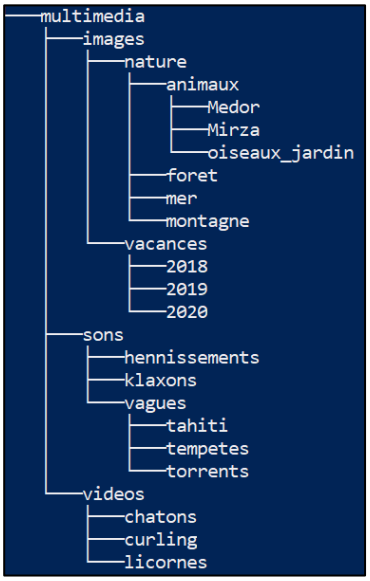
Un processus est l'objet dynamique associé à un programme en train de s'exécuter. Il peut y avoir plusieurs processus associés au même programme, par exemple on peut lancer simultanément plusieurs fenêtres d'un même navigateur web ou un programme peut avoir un processus correspondant à la fenêtre graphique et un processus correspondant à une impression papier).

Les systèmes d'exploitation grand public (de type UNIX , Windows, iOS, Android etc.) sont multi-tâches : l'utilisateur a l'impression que plusieurs processus peuvent s'exécuter en même temps (on imprime un document en écoutant de la musique et en surfant sur le web). Pourtant un processeur ne peut exécuter qu'un seul processus à la fois. Les différents processus en cours d'exécution sont donc "découpés" et traités successivement les uns après les autres par le processeur, petit morceau par petit morceau.

Le système d'exploitation doit donc changer régulièrement (des centaines de fois par seconde) de processus actif. Cela nécessite de garder en mémoire l'ensemble de la mémoire associé à chaque processus (le code qu'il exécute, les données, **le registre PC** qui lui permet de savoir où il en est de son exécution, etc.). Et donc l'OS doit recharger ce contexte à chaque fois que le processeur reprend les calculs d'un processus.

Un bloc = x micro secondes. En gris : le processus est actif. Le temps s'écoule vers la droite.																							
Processus 1 (impression)						■	■	■				■	■	■										
Processus 2 (musique)	■								■	■				■	■							■	■
Processus 3 (navigateur)		■	■								■					■	■	■	■				
Processus 4 (Python)				■	■															■	■		

3. Les fichiers.



Nous nous intéressons uniquement à la représentation *logique* des données (fichiers) pour un utilisateur d'un système de type `UNIX`.
Les fichiers sont des unités persistantes d'informations créées par un processus. Les fichiers sont gérés par l'OS.

3.1. Hiérarchie du stockage des données.

L'organisation logique des données dans un système `UNIX` est la suivante : les données sont rangées dans des ***fichiers,*** eux-mêmes organisés de façon hiérarchique, grâce à des ***répertoires***. Cette hiérarchie peut être représentée par une arborescence : le répertoire qui se trouve en haut de cette hiérarchie s'appelle le ***répertoire racine*** (ou ***racine : désigné par /***) et l'unique répertoire contenant un autre répertoire s'appelle le ***répertoire parent (ci-contre nature est parent de mer)***.

Remarque :
Sous Windows on peut avoir l'arborescence d'un fichier en ligne de commande (comme l'image ci-contre) grâce à la commande `tree`.
Sous Unix on installera tree au préalable (apt install tree).
https://www.tecmint.com/linux-tree-command-examples/

Pour désigner le chemin d'un répertoire on dispose de deux modes

- **absolu : /home/herve/multimedia/sons/vagues/tahiti**
- **relatif : si on est déjà dans le rép. multimedia on peut utiliser : ./nature/animaux/Mirza**

Au début du chemin, / indique que l'on part de la racine alors que ./ indique que l'on part du répertoire dans lequel on est (courant).

3.3. Les droits.

Sous UNIX, chaque fichier (et répertoire) est la **propriété** d'un **utilisateur** particulier ; par défaut il s'agit de l'utilisateur qui a créé le fichier. Les utilisateurs peuvent être réunis en **groupes**. Les droits sur les fichiers sont alors définis en fonctions de la "position" du demandeur par rapport au propriétaire du fichier : **propriétaire**, **faisant partie du groupe propriétaire**, **autre utilisateur**.

Sur un serveur de fichiers comme celui du lycée cela permet par exemple de cloisonner les dossiers de partage : les élèves de première appartiennent au groupe "premiere" et ont accès au dossier /Tous_groupes/premiere/
Mais ils n'ont pas accès au dossier /Tous_groupes/seconde/ qui est réservé aux membres du groupe "seconde".
En revanche ils ont accès – ainsi que les élèves de seconde – au dossier /Tous_groupes/eleves/ car tous les élèves sont membres du groupe "eleves".

Sur des machines partagées, cela permet aussi de protéger les fichiers sensibles (par exemple le cœur du système d'exploitation) et d'éviter qu'un utilisateur ne les lise, ne les modifie ou ne les exécute (par erreur ou volontairement). Sur les machines du lycée (certes sous Windows), les élèves ne peuvent ainsi pas exécuter certains programmes alors que les enseignants ou les administrateurs le peuvent.

Enfin, puisqu'il est nécéssaire que le propriétaire (ou administrateur) de la machine puisse tout contrôler, il existe un super-utilisateur : `root` qui peut créer des utilisateurs, modifier le propriétaire d'un fichier, supprimer les fichiers de n'importe quel autre utilisateur, exécuter des instructions sensibles qui sont réservées à lui seul etc.

Il y a trois types de droits (**lecture**, **écriture**, **exécution**, respectivement identifiés par les lettres **r**, **w** et **x** et les valeurs **4**, **2** et **1**) dont la signification est résumée dans le tableau ci-dessous :

	Fichier régulier	**Répertoire**
LECTURE r 4	Regarder le contenu	Lister le contenu
ECRITURE w 2	Modifier le contenu	Ajouter ou supprimer un élément
EXECUTION x 1	Exécuter	Passer à travers

III. OS libres VS propriétaires.

Il existe divers systèmes d'exploitation et tous n'apportent pas les mêmes solutions. Indépendamment des solutions apportées, il existe deux familles de systèmes : les systèmes **propriétaires (Windows, macOS …)** et les systèmes **libres (Linux, Android …)**.

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 (il est quand même prudent de connaître la licence sous laquelle le logiciel a été publié qui peut préciser ces droits et des obligations du type citer les auteurs originaux ou non, pouvoir fabriquer des logiciels propriétaires ou non, etc.).

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.