



Systemes d'Exploitation

INTRODUCTION ET GENERALITES

I. ARCHITECTURE DE L'ORDINATEUR

I.1. Architecture de l'ordinateur

Un ordinateur est composé d'une partie matérielle et d'une partie logicielle. La partie matérielle est composée de l'unité centrale et des périphériques. Cette unité centrale est composée d'un ou de plusieurs processeurs (UCT pour Central Process Unit) , d'une mémoire centrale (Random Access Memory RAM, et Read Only Memory ROM), des périphériques d'entrées (clavier, souris, scanner, micro, etc..), des périphériques de sorties (écran, baffles, imprimante, etc.), des périphérique d'entrées/sorties (les disques, les interfaces de connexion à des réseaux, etc.). La complexité du matériel implique la réalisation d'une couche logicielle qui gère celui-ci. Le Système d'Exploitation SE est un programme qui prend en compte tous ces composants, et les utilisent correctement de façon optimale.

Les logiciels : il existe différentes types de logiciels.

- les micro-logiciels qui sont fournis par le fabricant du matériel. Et pour un ordinateur, il s'agit du BIOS (Basic Input Outup System) logé dans la mémoire ROM
- les programmes systèmes qui permettent le fonctionnement de l'ordinateur. Parmi ceux-ci, le système d'exploitation. Le programme « système d'exploitation » est le programme fondamental des programmes systèmes. Il contrôle les ressources de l'ordinateur et fournit la

base sur laquelle seront construits les programmes d'application. Le SE soustrait le matériel au regard du programmeur et offre une présentation agréable des fichiers.

- les programmes d'application des utilisateurs qui répondent aux besoins de ces derniers sous forme d'application. La **Figure1** montre l'architecture en couche d'un ordinateur.

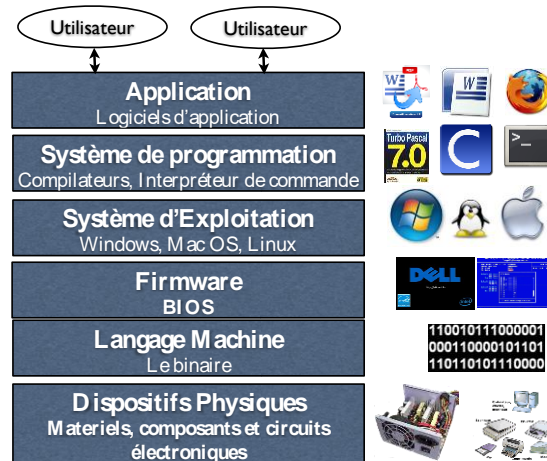


Figure1 : architecture en couche d'un ordinateur

I.2. Définition

Un Système d'Exploitation est un logiciel qui se charge de la gestion des ressources matérielles comme la RAM, le processeur et les périphériques, et qui s'occupe également de la gestion des ressources logicielles (programme, fichiers etc.) à travers des processus.

Un SE a ainsi deux objectifs principaux :

- **Présentation** : Il propose à l'utilisateur une abstraction plus simple et plus agréable que le matériel : une machine virtuelle
- **Gestion** : il ordonne et contrôle l'allocation des processeurs, des mémoires, des icônes et fenêtres, des périphériques, des réseaux entre les programmes qui les utilisent. Il assiste les programmes utilisateurs. Il protège les utilisateurs dans le cas d'usage partagé.

II. ELEMENTS DE BASE D'UN SYSTEME D'EXPLOITATION

Les principales fonctions assurées par un SE sont les suivantes :

La gestion des processus: un processus est un programme en cours d'exécution. Le système d'exploitation doit gérer l'allocation des ressources aux processus. Plusieurs processus peuvent exister simultanément, le SE doit être en mesure de gérer l'allocation des processus au processeur en proposant un environnement virtuel approprié dans lequel il peut évoluer en toute sécurité. La **Figure2** montre l'ordonnancement des processus par le SE dans une file d'attente du processeur.

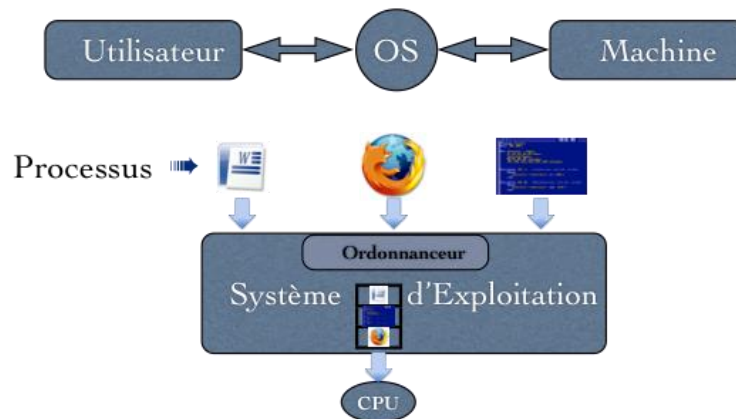


Figure2 : La file d'attente du processeurs

La gestion de la mémoire: le SE doit gérer l'allocation de la mémoire en déterminant les emplacements de mémoire auxquels peut accéder un processus. Elle doit également gérer les espaces libres et trouver des mécanismes efficaces pour optimiser l'occupation de la RAM. La Figure3 montre qu'avant qu'un processus ne soit exécuté par le processeur, il doit d'abord être présent dans la RAM.

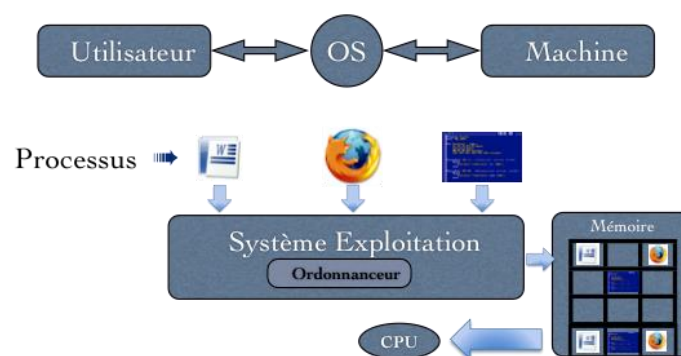


Figure3 : Le SE et l'allocation de la RAM

La gestion du système de fichier: les fichiers sont gérés par le SE. Leur structure, nommage, accès, utilisation, protection, et implantation sont des éléments majeurs de la conception d'un SE. Généralement, la partie du SE qui gère les fichiers est appelée le **système de gestion des**

fichiers (SGF). L'organisation des fichiers s'effectue selon une structure d'arbre comme le montre la **Figure4**.

- UASZ est la racine est désignée par /
- les nœuds sont les répertoires (dossier dans les systèmes Windows) non vides
- les feuilles sont les répertoires "vides" et les fichiers.

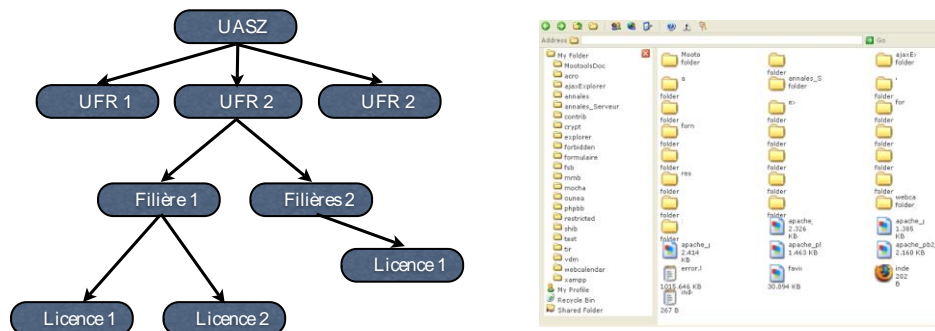


Figure4 : L'arborescence du système de Fichier

La gestion des périphériques: de nombreux types de périphériques existent (clavier, écran, imprimante etc...). La gestion de ces périphériques est effectuée par le SE. Ainsi, pour qu'un périphérique communique avec l'ordinateur, ses pilotes doivent être compatibles et intégrés ou installés sur le SE. Par exemple pour faire marcher correctement votre appareil photo branché sur votre ordinateur, en plus du Logitech (application), il doit être reconnu par l'ordinateur à travers ses pilotes intégrés dans le SE (voir **Figure4**).

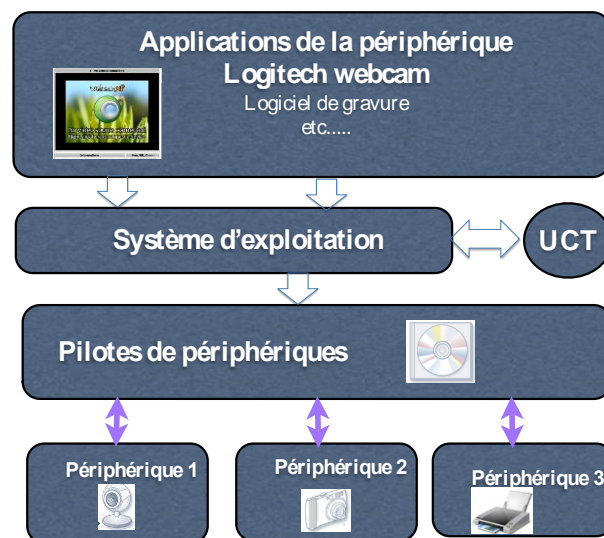


Figure4 : Le SE et les périphériques

III. HISTORIQUE DES SYSTEMES D'EXPLOITATION

Tout système d'exploitation dépend étroitement de l'architecture de l'ordinateur sur lequel il fonctionne. L'évolution des systèmes d'exploitation est en corrélation avec l'évolution technologique des ordinateurs. On distingue plusieurs générations d'ordinateurs.

La 1ère génération (1945 - 1955): les tubes à vide et les cartes enfichables (ou cartes perforées).

Il n'existait pas de système d'exploitation, ni de langages de programmation. L'élément de base de la technologie de fabrication de l'ordinateur est le tube à vide. A l'époque, un ordinateur remplissait une salle entière. Ces ordinateurs étaient d'une très grande lenteur et d'une très grande fragilité.

Il n'y avait pas de langage de programmation, la programmation se faisait en **langage machine absolu** souvent en basculant des tableaux d'interruption pour contrôler les fonctions de base de la machine.

Au début des années 1950, avec l'introduction des cartes perforées, il était devenu possible d'écrire ses programmes sur des cartes et de les faire lire par la machine au lieu d'utiliser les tableaux d'interruption.

La 2ème génération (1955 - 1965): les transistors et systèmes par lots.

Le passage aux **transistors** rendait les ordinateurs plus **fiables** et plus petits en termes de taille. Les utilisations principales étaient le **calcul scientifique** et **l'ingénierie** (équations différentielles).

On note aussi l'apparition des **premiers systèmes d'exploitation (SE)** de type **traitement par lot**: FMS (Fortran Monitor System) et IBSYS (IBM 7094). On note également l'apparition des langages de programmation comme le **FORTRAN** et le langage d'**assemblage**. Pour soumettre un job, le programmeur commençait par écrire son programme sur papier (en FORTRAN ou en assembleur), puis il le codait sur des cartes perforées. Il apportait ensuite son paquet de jobs et le confiait à l'opérateur présent. Quand l'ordinateur finissait un job, un opérateur récupérait sur une imprimante la trace d'exécution correspondante et le stockait quelques parts afin que le programmeur la récupère.

La 3ème génération (1965 - 1980) : les circuits intégrés et la multiprogrammation.

Amélioration des coûts et des performances (circuits intégrés). Les circuits intégrés ont contribué à la réduction des taille des ordinateurs: on parle de mini-ordinateurs. Une des caractéristiques des systèmes d'exploitation de troisième génération (OS/360, Freed's Books..)

est la technique appelée **spoulage** (de spool, Simultaneous Peripheral Operation On Line) qui est la capacité à stocker sur disques les jobs écrits sur cartes au fur et à mesure de leur arrivée. Ainsi dès qu'un job était terminé, le système d'exploitation pouvait en charger un nouveau depuis le disque et l'exécuter sur la partition libérée. Ces systèmes n'étaient encore que des systèmes batch. On note l'apparition de la **multiprogrammation** qui donne la possibilité d'avoir plusieurs programmes qui coexistent en RAM. Une variante de la multiprogrammation est du **partage de temps (time sharing)**, où les programmes de plusieurs utilisateurs peuvent être traités. Enfin des langages de programmation plus évolués (de haut niveau) comme le PASCAL et le C ont vu le jour.

La 4ème génération (1980 - 1990) : les Large Scale Integration (circuit à large échelle d'intégration) et les ordinateurs personnels.

Ils sont dûs au développement des **circuits LSI** (Large Scale Integration) contenant des centaines de transistors au cm^2 . Cette capacité d'intégration suit la loi empirique de Moore : *le nombre de transistors dans un circuit intégré double tous les 18 mois*. C'est l'ère des **micro-ordinateurs** (ordinateur personnel) qui ont à peu près la **même architecture que les mini-ordinateurs**. On assiste à l'apparition des systèmes d'exploitation comme **MS-DOS** (Microsoft Inc.) et **UNIX**. MS-DOS intègre petit à petit des concepts riches d'UNIX et de MULTICS. Dans le milieu des années 80, on voit l'**apparition de réseaux d'ordinateurs individuels** qui fonctionnent sous des systèmes d'exploitation en réseau ou des systèmes d'exploitation distribués. On peut encore noter l'apparition des langages de programmation plus évolués comme le langage C++ etc.

La 5ème génération (1990 - ????) : les Very Large Scale Integration (circuit à haute échelle d'intégration) les ordinateurs personnels portables et de poche.

L'évolution technologique suit la loi de Moore (voir **Figure5**). Ce qui réduit considérablement la taille des ordinateurs : ordinateurs portables, ordinateurs de poche (Parsonnal Digital Assistant PDA)

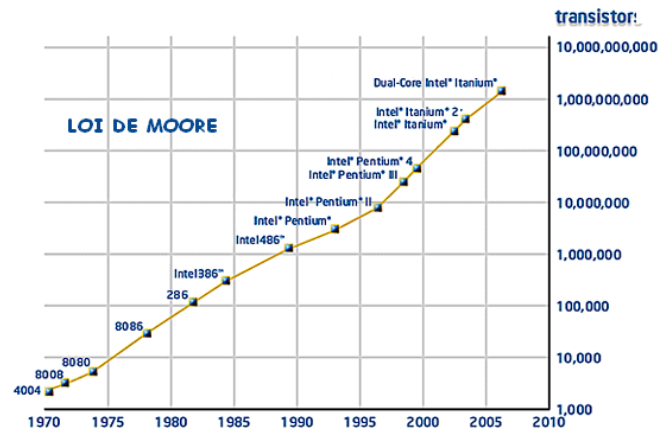


Figure5 : Loi de Moore

On note l'arrivée des systèmes d'exploitation de type « micro-noyau » qui sont modulaires (un module par fonction) ; ils peuvent être réalisés avec plus ou moins de modules et donc adaptables à de très petites machines (PDA et PIC).

IV. LES TYPES DE SYSTEME D'EXPLOITATION

On distingue plusieurs types de SE:

-Les SE en mode traitement par lots

Dans les premiers temps de l'informatique, les utilisateurs bénéficiaient rarement d'un accès direct aux ordinateurs. Les entrées, qu'il s'agisse de programmes ou données étaient préparées sur le support d'entrée, le plus souvent une bande ou carte perforée. Ainsi, leurs travaux sont envoyés en traitement sous forme de séquence par lot, il n'y a pas d'interactions entre le travail et l'utilisateur. Chaque tâche disposait d'un contrôle total sur la machine jusqu'à ce qu'elle se termine. C'est ce type de machine que gère un système d'exploitation en mode traitement par lots (batch mode).

Pour les systèmes par lots en multiprogrammation (multiprogram batch system), les tâches en cours d'exécution sont lues à partir d'un ensemble de tâches stockées sur le disque. Lorsqu'une tâche n'est pas en mesure de s'exécuter car elle attend la fin d'une opération d'E/S, l'exécution d'une autre tâche peut être autorisée.

-Les SE conversationnels

Le traitement est interactif entre l'utilisateur et le travail grâce au moniteur. Il a l'accès directement à l'ordinateur. Il reçoit des messages et des aides dans la recherche des erreurs (debugging). Le temps de réponse est plus court.

-Les SE à temps partagé

Les systèmes d'exploitation à temps partagé (time Sharing) tout comme les SE conversationnels autorisent des interactions entre l'utilisateur et le processus. Dans les systèmes par lots, toutes les données sont fournies au moment où le programme est saisi. Cependant, pour les programmes devant interagir avec l'utilisateur, le système d'exploitation doit prendre en charge un environnement permettant aux programmes de répondre aux saisies de l'utilisateur dans un laps de temps relativement raisonnable. En temps partagé, le système assure également le travail simultané de plusieurs utilisateurs, chacun étant directement lié avec son traitement. Les ressources sont partagées entre les utilisateurs. Chaque utilisateur a l'illusion de travailler seul avec toute la machine à son entière disposition alors qu'il a le contrôle seulement d'un quantum de temps. Le système d'exploitation doit non seulement partager des ressources entre les divers processus, mais il doit également agir comme si les processus s'exécutaient simultanément : **pseudo-parallélisme**. Pour cela, il bascule très rapidement d'un processus actif à l'autre.

-Les SE en temps réel

Un système d'exploitation à temps réel est conçu pour prendre en charge l'exécution de tâches dans le cadre de contraintes liées à l'horloge. En règle générale, les utilisateurs souhaitent que leur système exécute leur programme le plus rapidement possible, sans qu'aucun délai ne soit précisé. Dans les systèmes à temps réel, la justesse de l'exécution d'une tâche dépend du moment où surviennent les événements externes. Le système doit garantir que la tâche peut être exécutée dans un délai spécifié. L'impossibilité de réagir à temps réel entraîne une déformation de la procédure de calcul, ce qui peut avoir comme résultat une catastrophe. Les applications sont souvent privées (industrie, télécom, médecine etc....).

-Les SE repart

Avec un système d'exploitation en réseau, les ressources de chaque machine du réseau sont gérées par le système d'exploitation de cette machine. La prise en charge des systèmes d'exploitation simplifie les communications entre machines. Avec un système d'exploitation réparti, les systèmes d'exploitation de toutes les machines fonctionnent conjointement pour gérer les ressources collectives du réseau. Un seul système d'exploitation collectif gère les ressources du réseau qui sont fournies par chaque ordinateur du réseau ou nœud.

V. LA STRUCTURE D'UN SE

Le système d'exploitation est constitué de deux parties :

Une partie résidente appelée **noyau**

Une partie **transitaire** appelée extension du SE

Le noyau est constitué des programmes et des données en mémoire et qui y reste pendant tout le temps de fonctionnement du SE. Il est constitué entre autre des gestionnaires d'opérations de base, du processeur et des processus de la mémoire, des fichiers, des entrées/sorties, des périphériques, de la sécurité.

L'extension du SE est constituée de logiciels appelés suivant leur nécessité (chargés au besoin et peuvent être déchargés). Ce sont essentiellement les utilitaires comme l'interprétation de commande, les programmes d'interaction avec le système de programmation.