



**SYSTEME D'EXPLOITATION**  
**Notes de cours destinées aux étudiants**  
**de BAC II (Informatique de Gestion)**  
**à l'Université Lumière de BUJUMBURA de Goma.**

Appartenant à l'Étudiant : .....

1<sup>ère</sup> édition, Août 2022

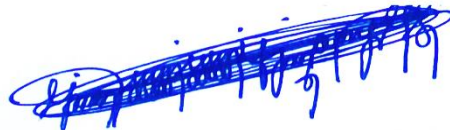
## Avertissements

Ce support de cours est soumis aux droits d'auteur et n'est donc pas dans le domaine public. Sa reproduction est cependant autorisée à condition de respecter les conditions suivantes :

- Φ Si ce document est reproduit pour les besoins personnels du reproducteur, toute forme de reproduction (totale ou partielle) est autorisée à la condition de citer l'auteur.
- Φ Si ce document est reproduit dans le but d'être distribué à des tierces personnes, il devra être reproduit dans son intégralité sans aucune modification. Cette notice de copyright devra donc être présente. Qui plus est, il ne devra pas être vendu.
- Φ Nonobstant, dans le seul cas d'un enseignement gratuit, une participation aux frais de reproduction pourra être demandée, mais elle ne pourra être supérieure au prix du papier et de l'encre composant le document. ! Toute reproduction sortant du cadre précisé ci-dessus est interdite sans accord préalable écrit de l'auteur.

Toute reproduction sortant du cadre précisé ci-dessus est proscrite sans accord préalable écrit de l'auteur.

**Copyright © 2022 Drs. Ir. Jacques MUDUMBI A; all rights reserved.**



## Table de matières

Avertissements .....	2
Table de matières.....	3
Liminaires.....	5
Contenu.....	7
Bibliographie sélective.....	8
INTRODUCTION.....	9
CHAPITRE I : GENERALITES SUR LES SYSTEMES D'EXPLOITATIONS .....	10
I.1. Vue générale d'un système d'exploitation .....	10
I.2. LES FONCTIONS D'UN SYSTEME D'EXPLOITATION.....	11
I.3. HISTORIQUE DES PRINCIPAUX SYSTEMES D'EXPLOITATION .....	12
I.3.1. INTRODUCTION A MICROSOFT WINDOWS.....	12
I.3.2. INTRODUCTION A UNIX.....	25
I.3.3. INTRODUCTION A MacOS.....	32
I.4. LE PARALLELISME DES SYSTÈMES D'EXPLOITATION (WINDOWS, LINUX ET MAC OS) SE DESSINE À TRAVERS LE TABLEAU COMPARATIF CI-DESSOUS .....	37
I.5. PREDILECTION DE L'OS PAR L'UTILISATEUR.....	39
I.6. EVOLUTION DES SYSTEMES D'EXPLOITATION .....	40
CHAPITRE II : STRUCTURES ET ARCHITECTURES DES SYSTEMES D'EXPLOITATION .....	42
II.1. COMPOSANTS DU SYSTEME D'EXPLOITATION.....	43
II.2. LES TYPES DE SYSTEME D'EXPLOITATION .....	44
II.3. CLASSES DES SYSTEMES D'EXPLOITATION.....	45
II.4. NOTION SUR LE NOYAU D'UN SYSTEME D'EXPLOITATION .....	46
II.4.1. DEFINITION .....	46
II.4.2. ARCHITECTURES DE DIFFERENTS TYPES DE NOYAUX .....	48
II.5. PERTINENCE DU SYSTEME D'EXPLOITATION SUR LA VIRTUALISATION.....	52
II.6. L'ARCHITECTURE CLIENT/SERVEUR .....	54
II.7. DEMARRAGE DU SYSTEME D'EXPLOITATION.....	54
CHAPITRE III: GESTION DES RESSOURCES INFORMATIQUES: LES MEMOIRES, LES PERIPHERIQUES ET LES FICHIERS .....	56
III.1. DEFINITION .....	56

III.2. GESTION DE LA MEMOIRE .....	56
III.2.1. Gestion de la mémoire pour systèmes monotâches.....	57
III.2.2. Gestion de la mémoire pour systèmes multitâches .....	58
III.2.3. La pagination.....	59
III.2.4. La segmentation.....	62
III.2.5. Segmentation avec pagination.....	63
III.3. GESTION DES PERIPHERIQUES (devmgmt.msc).....	63
III.3.1. Notion .....	64
III.3.2. Organisation des dispositifs d'E/S.....	64
III.3.3. Contrôle des E/S.....	66
III.3.4. Ports d'E/S .....	66
III.3.5. Communication entre UC et E/S.....	66
III.3.6. Les pilotes des périphériques .....	67
III.3.7. Les périphériques .....	67
III.4. GESTION DES FICHIERS .....	69
III.4.1. Notion .....	69
III.4.2. Les fichiers .....	69
III.4.3. Répertoires, noms de fichiers et partitions .....	70
III.4.4. Type des objets du système de fichiers .....	70
III.4.5. Fonctions des systèmes de fichiers .....	71
III.4.6. Architecture du système de fichiers .....	71
III.4.7. Allocation .....	75
Conclusion .....	78

## Liminaires

### 1. Point de vue du cours et sa justification

*« Quel système d'exploitation*

*correspond le mieux à vos besoins personnels ?*

*Réponse : celui qui est déjà sur votre ordinateur.*

*Croyez-moi, personne n'a vraiment envie d'en installer un nouveau.»*

*(Dave Barry)*

Au tout début de l'informatique, les premiers ordinateurs sont apparus dans les années 40 et avaient des fonctionnalités très limitées, dépendantes de l'homme, et à des vitesses très lentes. Le logiciel assurait lui-même la gestion du matériel. Certes, chaque application gérait elle-même la mémoire, la communication avec les périphériques et bien d'autres choses. Le logiciel pouvait être démarré directement à l'allumage de l'ordinateur, sans charger de système d'exploitation. En conséquence, les logiciels n'étaient pas compatibles sur une large gamme de matériel et supportaient assez mal un changement de configuration. Mais avec le temps, les informaticiens ont inventé des techniques d'abstraction matérielle qui permettent à un programme de s'exécuter sur des ordinateurs avec des matériels très différents. Rendre les programmes indépendants du matériel a demandé de revoir l'organisation des logiciels, qui ont dû être découpés en plusieurs programmes séparés (les programmes systèmes gèrent la mémoire et les périphériques, les programmes applicatifs ou applications délèguent la gestion de la mémoire et des périphériques aux programmes systèmes).

Au demeurant, l'ensemble des programmes système porte le nom de système d'exploitation abrégé **SE** ou **OS**). Le rôle du système d'exploitation est d'être un intermédiaire entre les logiciels et le matériel. A vrai dire, pour qu'un ordinateur soit capable de faire fonctionner un programme informatique, la machine doit être en mesure d'effectuer un certain nombre d'opérations préparatoires afin d'assurer les échanges entre le processeur, la mémoire, et les ressources physiques (périphériques). De surcroît, le système d'exploitation, est chargé d'assurer la liaison entre les ressources matérielles, l'utilisateur et les applications. Ainsi lorsqu'un programme désire accéder à une ressource matérielle, il ne lui est pas nécessaire d'envoyer des informations spécifiques au périphérique, il lui suffit d'envoyer les informations au système d'exploitation, qui se charge de les transmettre au périphérique concerné via son pilote. En l'absence de pilotes il faudrait que chaque programme reconnaisse et prenne en compte la communication avec chaque type de périphérique !

Mais réflexion faite, le système d'exploitation (Operating System) est le programme qui relie l'utilisateur avec la machine. Il contrôle l'allocation et l'utilisation de la partie physique de l'ordinateur (Hardware), allant des actions les plus simples (démarrer l'ordinateur, reconnaître les entrées clavier ou souris, lire des fichiers...) aux plus complexes (lancement d'applications, utilisation de périphériques...). Non seulement mais aussi, il sert trois intentions distinctes : le système d'exploitation permet l'exploitation des périphériques matériels dont il coordonne et optimise l'utilisation; Il propose aux logiciels applicatifs des interfaces de programmation standardisées qui simplifient l'utilisation des matériels et réalise enfin différentes fonctions visant à assurer la fiabilité (tolérance aux pannes, isolation des fautes) et la sécurité informatique (traçabilité, confidentialité, intégrité et disponibilité).

C'est dire aussi que, en allumant l'ordinateur portable, on peut voir d'abord comment le système démarre. Selon le système d'exploitation choisi, **Windows, Linux ou macOS** démarre. Ce n'est que lorsque ce système, si important pour le fonctionnement de l'appareil, est chargé que vous pouvez vous mettre au travail et lancer des programmes, vous connecter à Internet ou ouvrir des fichiers. Toutefois, le système d'exploitation n'est pas responsable du démarrage de l'appareil. Il est précédé par le BIOS, qui est responsable du processus de démarrage proprement dit. Toutefois, le BIOS, n'a que des capacités très rudimentaires, qui sont principalement liées au processus de démarrage. Lors de ce dernier, le BIOS démarre également le système d'exploitation, qui prend ensuite en charge l'administration complète de l'ordinateur. Une grande partie de ces activités se déroulent en arrière-plan. Le logiciel très complet permet d'effectuer des réglages sur les appareils, d'installer et de désinstaller des programmes et d'exécuter des logiciels.

À bien considérer les choses, un système d'exploitation peut servir à coordonner l'utilisation du ou des processeur(s), et accorder un certain temps pour l'exécution de chaque processus ; à réserver de l'espace dans les mémoires pour les besoins des programmes et à organiser le contenu des disques durs ou d'autres mémoires de masse en fichiers et répertoires. Il peut aussi servir à créer l'image numérique qui contiendra les interfaces homme-machine des différents programmes et à l'envoyer au moniteur ainsi qu'à recevoir les manipulations effectuées par l'utilisateur *via* le clavier, la souris ou d'autres périphériques, et les transmettre aux différents programmes. Lorsque l'ordinateur démarre (Boot), il cherche le système d'exploitation et une fois qu'il l'a trouvé, il le charge dans la mémoire RAM. Il est alors apte à répondre aux demandes de l'utilisateur et des applications. Le système d'exploitation gère la mémoire : il sauvegarde, supprime et récupère des données, les affiche sur l'écran, les envoie aux périphériques qui en ont besoin (les imprimantes par exemple), il répond aux entrées clavier, reconnaît les clics de souris et il lit et écrit des données sur des

disques durs ou autres périphériques de stockage. C'est alors que ce cours a pour objectif de décrire les généralités des systèmes d'exploitation, expliquer la structure et l'architecture des systèmes d'exploitation voire la gestion des ressources matérielles ce qui signifie la transformation du matériel en une machine utilisable et l'optimisation de l'utilisation des ressources matérielles et logicielles.

## 2. Démarche pédagogique

Le cours repose sur une démarche d'actualisation des théories, pratiques et connaissances des prérequis sur les principes généraux de l'algorithmique. D'une part, ce cours fournit une vue d'ensemble de ce domaine d'études en prenant connaissance des concepts nécessaires pour le système d'exploitation. Bien plus, il distingue et explique les grandes notions des ressources matérielles et logicielles, notions avec lesquelles le SE réagit ainsi que celles de l'évolution de chaque OS voire la compréhension leurs différentes structures et architectures et de savoir-faire relatif à la résolution des problèmes complexes de gestion des ressources (les mémoires, les périphériques et les fichiers). Tout compte fait, le cours privilégie l'approche interactive.

## 3. Prérequis

Connaissances des principes généraux de l'algorithmiques et structures des données ;

## 4. Support pédagogique

Tableau blanc, Marqueur, Rétroprojecteur, Ordinateur et en plus l'étudiant dispose d'un fascicule rédigé par nous-même ;

## 5. Mode d'évaluation

L'examen final est écrit, il vise à vérifier la bonne maîtrise par l'étudiant des concepts et notions étudiées au cours. Les questions seront spécifiquement théoriques similaires à des notions essentielles développées à l'auditoire supposant une très bonne connaissance et compréhension de la matière. Il se compose d'un questionnaire avec 6 questions qui seront notés chacune sur 3 ou 4 points, le total des points à l'examen étant ensuite 24 qui sera divisé par 2 pour avoir une note d'ensemble sur 12, et 3 points pour l'étudiant assidu ayant participé au  $\frac{3}{4}$  des séances du cours, 5 points pour les travaux pratiques et l'interrogation pour obtenir une note d'ensemble sur 20.

## Contenu

Chapitre I : Généralités sur les systèmes d'exploitations ;

Chapitre II : Structures et architectures des systèmes d'exploitations ;

Chapitre III : Gestion des ressources : les mémoires, les périphériques et les fichiers ;

## Bibliographie sélective

- [1] Andrew Tanenbaum, *Systèmes d'exploitation*, Pearson Education France, 2008, 3<sup>e</sup> éd.
- [2] D.L. Galli, *Distributed Operating Systems*,
- [3] J.A. Montagnon, E.Pichat "Architecture des ordinateurs. Tome 1: le sous-système central" Masson 1986
- [4] J.C. Heudin, C. Panetto "Les architectures RISC " Dunod 1990
- [5] Joffroy Beauquier, Béatrice Bérard, Systèmes d'exploitation, Concepts et Algorithmes Ediscience
- [6] J.P. Vergus, G. Roucairol "Parallélisme, communications et synchronisation" Ed. CNRS 1985
- [7] J.-R. Tong Tong "Les communications sous Unix" Eyrolles 1991.
- [8] Laurent Bloch, *Les Systèmes d'exploitation des ordinateurs. Histoire, fonctionnement, enjeux*, Vuibert, 2003
- [9] Maurice Bach, *La conception du Système Unix*,
- [10] Michael Griffiths, Michel Vayssade, *Architectures des systèmes d'exploitation*, Hermes ,
- [11] P. Zanella, Y. Ligier "Architecture et technologie des ordinateurs" Dunod 1989
- [12] Peter J. Denning, *Pourquoi ne pas les innovations dans l'architecture informatique?*, en SIGARCH ACM Architecture des ordinateurs Nouvelles, vol. 8, No. 2, Avril 1980 pp. 4-7, ISSN 0163-5964.
- [13] Pour Brinch Hansen, *Le noyau d'un système Multiprogrammation*, en *Communications de l'ACM*, vol. 13, n° 4, Avril 1970, p. 238-241, ISSN 0001-0782.
- [14] R. Moreau "Ainsi naquit l'Informatique" Dunod 1987
- [15] S. Krakowiak "Principes des systèmes d'exploitation des ordinateurs" Dunod 1987



## INTRODUCTION

A priori, la plupart des systèmes d'exploitation sont conçus graphiquement et sont axés sur la convivialité. Les fonctions les plus importantes doivent être accessibles aux utilisateurs en quelques clics de souris ; la navigation dans le menu doit être aussi intuitive que possible. Cependant, cela n'a pas toujours été le cas. Au tout début, lorsque les premiers ordinateurs ont été utilisés à des fins scientifiques et militaires etc; il n'y avait pas alors de système d'exploitation. À cette époque, tout se faisait directement avec la machine elle-même, il n'y avait pas encore de système intermédiaire. Mais au moment où les premiers PC sont apparus, c'est-à-dire des ordinateurs qui n'avaient plus la taille d'une armoire, les systèmes d'exploitation étaient également des équipements standards.

Les premiers systèmes d'exploitation ne disposaient pas encore d'une interface utilisateur graphique. Ils fonctionnaient plutôt avec une ligne de commande, que vous pouvez encore trouver aujourd'hui dans le **Windows PowerShell**. Pour chaque action, qu'il s'agissait d'ouvrir un fichier, de naviguer dans la structure des dossiers ou de lancer un programme, vous deviez connaître et taper la commande correcte. Ce n'est que lorsque les ordinateurs ont pu être utilisés à domicile que les interfaces utilisateur plus simples sont devenues monnaie courante. Les deux sociétés Microsoft (avec Windows) et Apple (avec macOS) en sont à l'origine.

Un ordinateur sans logiciel ne sert à rien. C'est le logiciel qui donne à cet ensemble électronique la capacité de traiter les informations. Pour qu'un ordinateur soit capable de fonctionner il doit avoir un programme informatique (appelé parfois application ou logiciel), la machine doit être en mesure d'effectuer un certain nombre d'opérations préparatoires afin d'assurer les échanges entre le processeur, la mémoire et les ressources physiques (périphériques).

Le système d'exploitation noté S.E. ou O.S (abréviations du terme Anglais: Operating system) est chargé d'assurer la liaison entre les ressources matériels, l'utilisateur et les applications (traitement de texte, jeu vidéo, tableau,...) ainsi lorsqu'un programme désire accéder à une ressource matérielle, il ne lui est pas nécessaire d'envoyer des informations spécifiques au périphérique mais il lui suffit d'envoyer les informations au S.E. qui se charge de les transmettre aux périphériques concernés via son pilote (Un **pilote informatique**, souvent abrégé en **pilote**, est un programme informatique destiné à permettre à un autre programme (souvent un système d'exploitation) d'interagir avec un périphérique.). En l'absence de pilote, il faudrait que chaque programme reconnaisse un autre programme en tenant compte de la communication avec chaque type de périphérique.

## CHAPITRE I : GENERALITES SUR LES SYSTEMES D'EXPLOITATIONS

### I.1. Vue générale d'un système d'exploitation

Utiliser un ordinateur, c'est utiliser le système d'exploitation qui est dans son PC. Cependant posons-nous la question de s'avoir, **Qu'est-ce qu'un système d'exploitation?**

**Un système d'exploitation (noté SE ou OS pour Operating System)** est un ensemble de programmes responsables de la liaison entre les ressources matérielles (disque, mémoire, processeur, ...) d'un ordinateur et les applications de l'utilisateur (traitement de texte, Internet, jeu vidéo...). C'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. **Autrement dit le système d'exploitation** est un programme qui sert d'interface entre un utilisateur et un ordinateur.



Figure N° 1.1.0 : Vue générale d'un système d'exploitation

## I.2. LES FONCTIONS D'UN SYSTEME D'EXPLOITATION

Etant donné que le système d'exploitation 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 et protège les utilisateurs dans le cas d'usage partagé. Il regroupe certaines fonctions comme:

- ☑ **La Gestion du processeur:** Un processus est l'entité créée par le système d'exploitation pour l'exécution d'un programme. Un programme est une entité passive qui réside en mémoire. Le processus, en revanche, est placé instruction par instruction dans les registres de l'UC et exécuté. La principale tâche du système d'exploitation concerne l'allocation du processeur aux processus. Il s'agit de décider quel processus s'exécute à un moment donné, à quel moment interrompre le processus, quel sera le suivant, et de quoi il a besoin comme ressources pour son exécution.
- ☑ **La Gestion de la mémoire centrale:** La mémoire centrale est un espace de taille important organisé en mots (ensembles d'octets) et destinée à accueillir les données à traiter par l'unité centrale. L'unité centrale charge les instructions à exécuter dans les registres du processeur à partir d'adresses en mémoire centrale. De même après exécution, les résultats sont placés en mémoire centrale. Il s'agit ici de gérer l'allocation de cette mémoire aux programmes (attribution, libération de mémoire), les règles d'adressage et de veiller à ce que les programmes en mémoire ne puissent pas interférer entre eux. En cas d'insuffisance de mémoire physique, le système d'exploitation peut créer une zone mémoire sur le disque dur, appelée «mémoire virtuelle» qui permet de faire fonctionner des applications nécessitant plus de mémoire qu'il n'y a de mémoire vive disponible sur le système. En contrepartie cette mémoire est beaucoup plus lente.
- ☑ **La Gestion des entrées/sorties:** le système d'exploitation permet d'unifier et de contrôler l'accès des programmes aux ressources matérielles par l'intermédiaire des pilotes (appelés également gestionnaires de périphériques ou gestionnaires d'entrée/sortie).
- ☑ **La Gestion de l'exécution des applications:** le système d'exploitation est chargé de la bonne exécution des applications en leur affectant les ressources nécessaires à leur bon fonctionnement. Il permet à ce titre de «tuer» une application ne répondant plus correctement, ou de suspendre son exécution face à un déroutement (erreur dans un programme ex. division par zéro ou adresse mémoire inexistante) ou une interruption (signalé provoqué par un périphérique d'E/S).
- ☑ **La Gestion des droits:** le système d'exploitation est chargé de la sécurité liée à l'exécution des programmes en garantissant que les ressources ne sont utilisées que par les programmes et les utilisateurs possédant les droits adéquats.

- ☑ **La Gestion des fichiers:** le système d'exploitation gère la lecture et l'écriture dans le système de fichiers et les droits d'accès aux fichiers par les utilisateurs et les applications. Elle permet le stockage à long terme des données et programmes sur la mémoire de masse.
- ☑ **La Gestion des informations:** le système d'exploitation fournit un certain nombre d'indicateurs permettant de diagnostiquer le bon fonctionnement de la machine. *Bref, e système d'exploitation fournit un certain nombre d'indicateurs permettant de diagnostiquer le bon fonctionnement de la machine.*
- ☑ **La Gestion des protocoles réseau** (TCP/IP, IPX, etc.).

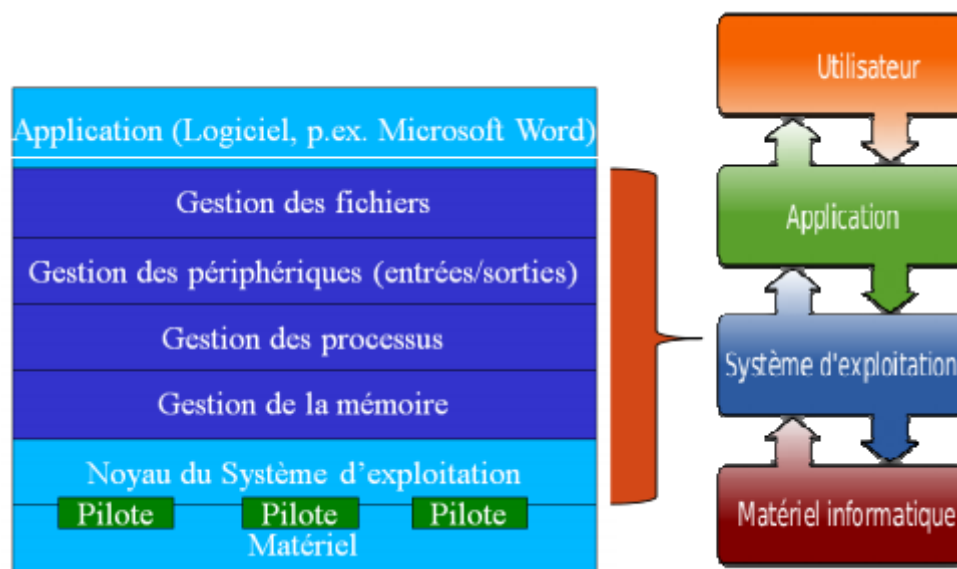


Figure N° 1.1.1 : Fonctions d'un système d'exploitation

## I.3. HISTORIQUE DES PRINCIPAUX SYSTEMES D'EXPLOITATION

### I. 3.1. INTRODUCTION A MICROSOFT WINDOWS



**Windows** est un système d'exploitation imaginé et développé par **Bill Gates**, PDG de **Microsoft**, depuis le milieu des années 1980. Ce système d'exploitation a été développé pour les ordinateurs personnels de bureau et portable. Depuis cette période, plusieurs versions ont été développées. Au départ, Windows a été développé sur les PC conçus par IBM et ses clones conçus par Microsoft. L'interface n'existait pas sous forme graphique à l'époque, tout fonctionnait en « lignes de commande ». Un peu plus tard, Windows 1 fait son apparition et propose pour la première fois une interface graphique pour les PC. A ce moment-là, le système n'intégrait que des opérations mono tâche c'est-à-dire que le système était lancé sous forme d'un simple programme reposant sur le système préexistant : MS-DOS. Certaines fonctions que l'on connaît

aujourd'hui étaient assurées par Windows 1 comme le tracking de la souris sur l'écran ou le multi fenêtrage qui ne faisait que séparer l'écran en deux sans superpositions. Windows 2 qui succéda au Windows 1 ajoutait la superposition de fenêtres mais la résolution était nettement plus faible qu'aujourd'hui. Windows 3, inspiré du projet Alto de Xerox, puis de Lisa et du Macintosh de Apple, propose l'architecture la plus proche de celle que nous connaissons aujourd'hui c'est-à-dire que Windows prend la forme d'une interface indépendante de l'ordinateur composée de plusieurs sous programmes. Cela lui donnera enfin son nom de Windows. Windows 3 utilisa la métaphore du bureau avec des fichiers, des dossiers et une corbeille. Cependant cela arriva après le lancement du Macintosh de Apple en 1990. A partir du milieu des années 1990, Windows prend sa forme définitive avec un menu « démarrer », ses fichiers, ses dossiers et sa corbeille. Plusieurs logiciels sont livrés avec le système, ce qui imposa les normes spécifiques au système tels que les fichiers en « .exe » ou « .doc » par exemple.

À cette époque les systèmes (ordinateurs) n'avaient ni la capacité ni la nécessité d'utiliser des systèmes d'exploitation pour mini ou grands ordinateurs. Les premiers systèmes d'exploitation appelés moniteurs offraient seulement des fonctionnalités très basiques, et étaient chargés le plus souvent depuis de la mémoire morte. CP/M fut l'un des pionniers en matière de système d'exploitation installé sur un disque (et non sur mémoire morte). Ce système d'exploitation influença largement la conception de PC-DOS. Ce dernier, choisi par IBM comme système d'exploitation pour ses premiers PC, fit de Microsoft l'une des compagnies les plus rentables au monde. Les principales alternatives durant les années 1980 sur le marché des systèmes d'exploitation furent Mac OS en 1984, AmigaOS et Atarien 1985 et OS/2 en 1987.

Ensuite, la fin des années 1990 a vu les exigences des utilisateurs s'accroître en termes de fiabilité, de sécurité et de convivialité de l'interface graphique. C'est ainsi que Microsoft a opéré une migration de tous ses systèmes d'exploitation, en proposant ces différents types de systèmes qui se sépare en 2 parties, les systèmes pour amateurs et pour professionnels. Au début des années 2000, Microsoft lance son fameux Windows XP qui fut un énorme succès grâce à son ergonomie et sa simplicité. Cependant, cette version mit en lumière des lacunes liées à la sécurité du système. Sa stabilité en fera un système toujours en activité 10 ans après. Windows Vista succéda à Windows XP mais ne trouvera pas un grand taux d'adoption en raison de sa lenteur et de son instabilité. On entend par stabilité d'un système d'exploitation, la capacité à rester actif sans que le système ne lâche en affichant un « blue screen ».

Il faudra attendre Windows 7, sorti en 2009, qui sera adopté par beaucoup d'utilisateurs. Il est encore en activité aujourd'hui.

Windows 8 sera lancé en 2012 mais ne sera pas plébiscité par le public du fait que son architecture soit complètement différente des versions précédentes. Le menu « démarrer » n'existe plus. Ce flop est dû au fait que cette version a été trop pensée pour le tactile. Il faudra attendre la dernière version sortie en 2016, Windows 10, pour voir le menu « démarrer » faire sa réapparition. Pour pallier aux lacunes de sécurité, Microsoft a mis en place un système de mise à jour permanente et obligatoire tous les six mois.

Voici un schéma non exhaustif retraçant globalement l'apparition des principaux systèmes de type WINDOWS:

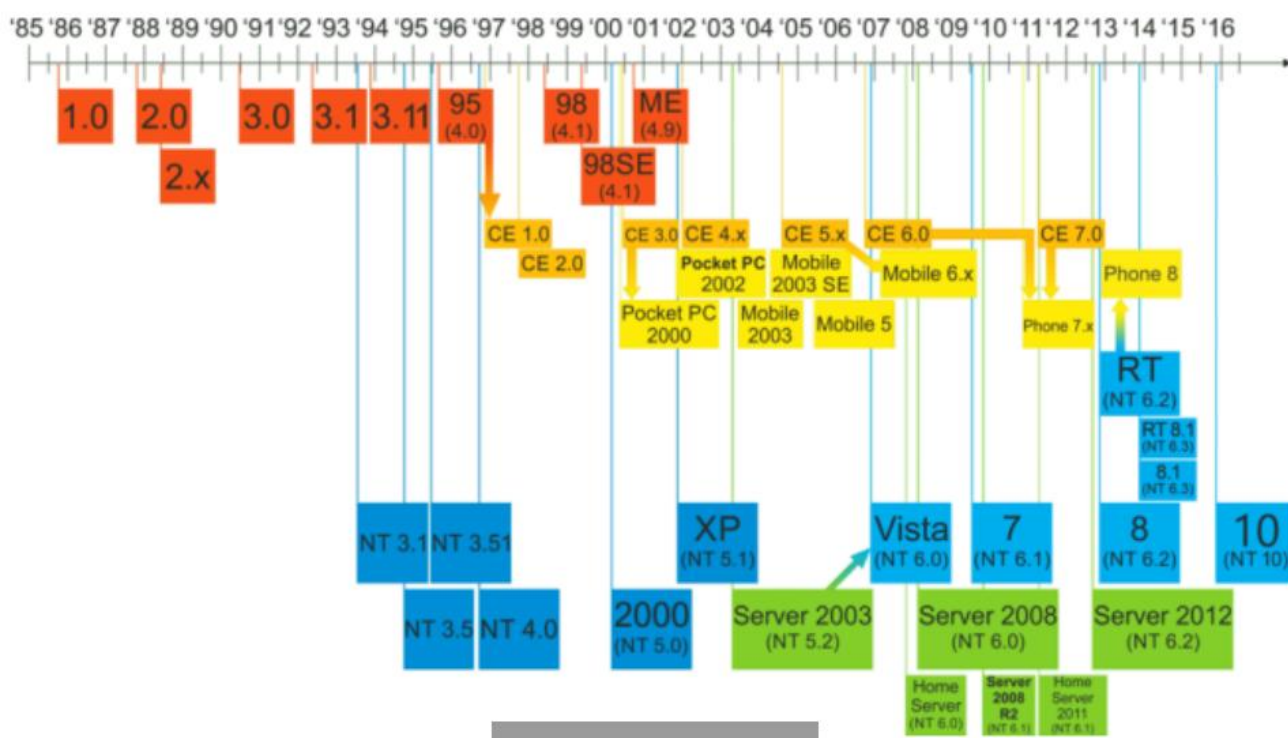


Figure N° 1.1.2: Evolution des systèmes d'exploitation de type Windows

## 1) INTERFACES DES SYSTEMES D'EXPLOITATION DE MICROSOFT

La famille Windows détient le quasi-monopole sur les ordinateurs personnels, avec plus de 90 % de part de marché depuis 15 ans, tandis que les parts de marché des systèmes d'exploitation Unix sont de presque 50% sur les serveurs. Système d'exploitation et base matérielle. **Windows** est à la fois un **système d'exploitation** et une **interface graphique** (GUI- Graphic User Interface).

Il existe en plusieurs versions Windows 1, 2, 3.X, 95, 98, Me (millénium), Windows NT, 2000, XP, Vista, Windows 7, 8 et 10. Nous ne citons ici que les versions pour particuliers sans parler des versions spécialisés pour les serveurs qui sont en général de plus gros ordinateurs configurés pour la gestion des réseaux.

Windows évolue régulièrement pour devenir à la fois de plus en plus performant et convivial, ses versions successives demandent pour tourner dans nos PC des processeurs de plus en plus rapides ainsi que des mémoires et des disques de plus en plus volumineux, le cout du matériel va heureusement en diminuant. Ceci dit, ceux qui disposent d'un matériel ancien auront parfois avantage à n'est pas se précipiter sur les dernières versions des logiciels et en particulier de Windows. Les améliorations apportées ne compenseront sans doute pas la perte de vitesse qu'entraînent, sur du matériel ancien, ces logiciels de plus en plus gourmands en ressources matérielles.

En gros, d'une version à l'autre, c'est surtout le fonctionnement interne du système d'exploitation qui change. Les modifications de l'interface restent relativement réduites, si bien qu'il n'est pas trop difficile de s'adapter aux versions successives.



*Figure N° 1.1.3: Interfaces de Windows 1.0 à Windows 10*

### **Windows 1**

Il y a plus de 30 ans, le 20 novembre 1985, Microsoft présentait Windows 1.01. Il s'agissait plus d'une interface graphique que d'un système. Quoi qu'il en soit, il n'était pas encore question de bouton ou de menu démarrer.

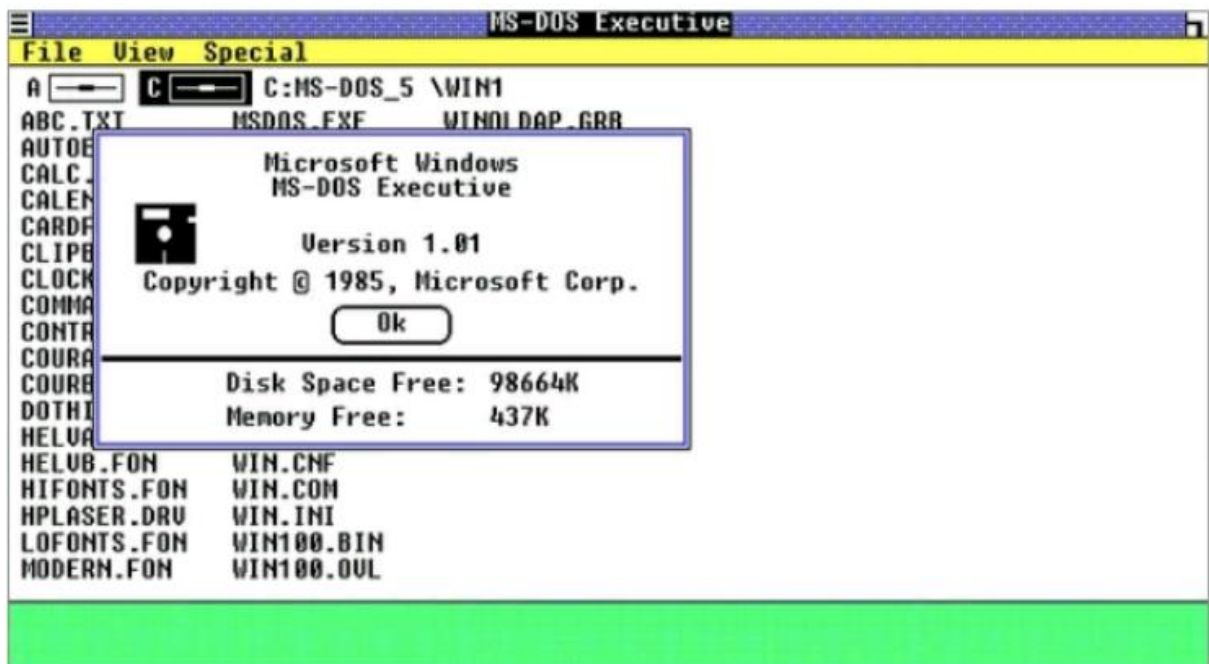


Figure N° 1.1.4: Interface de Windows 1

## Windows 2

Pas de dépaysement avec Windows 1 mais les fenêtres peuvent maintenant se superposer et le principe de réduction/agrandissement fait son entrée.

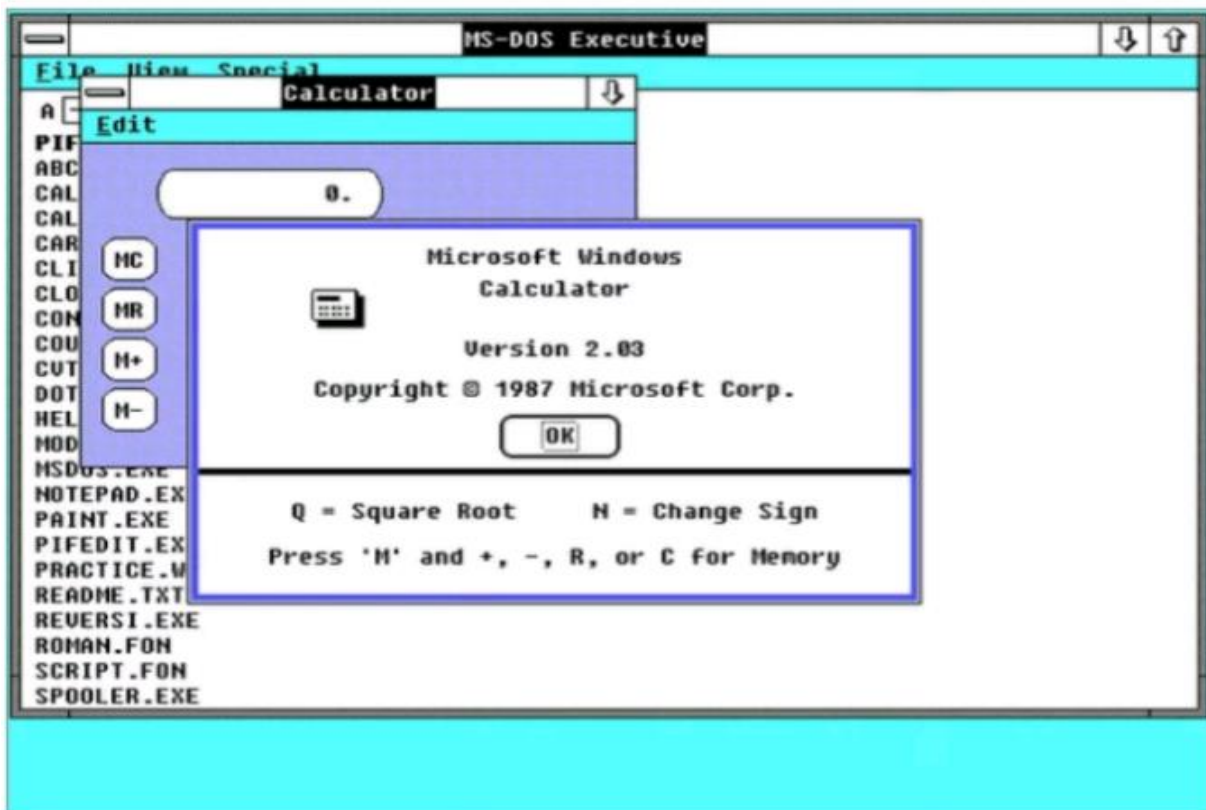


Figure N° 1.1.5: Interface de Windows 2



### Windows 3.x

Au début des années 90 apparaît la gamme des Windows 3.x, le design s'affine. On voit ici le gestionnaire de programmes de Windows 3.1.

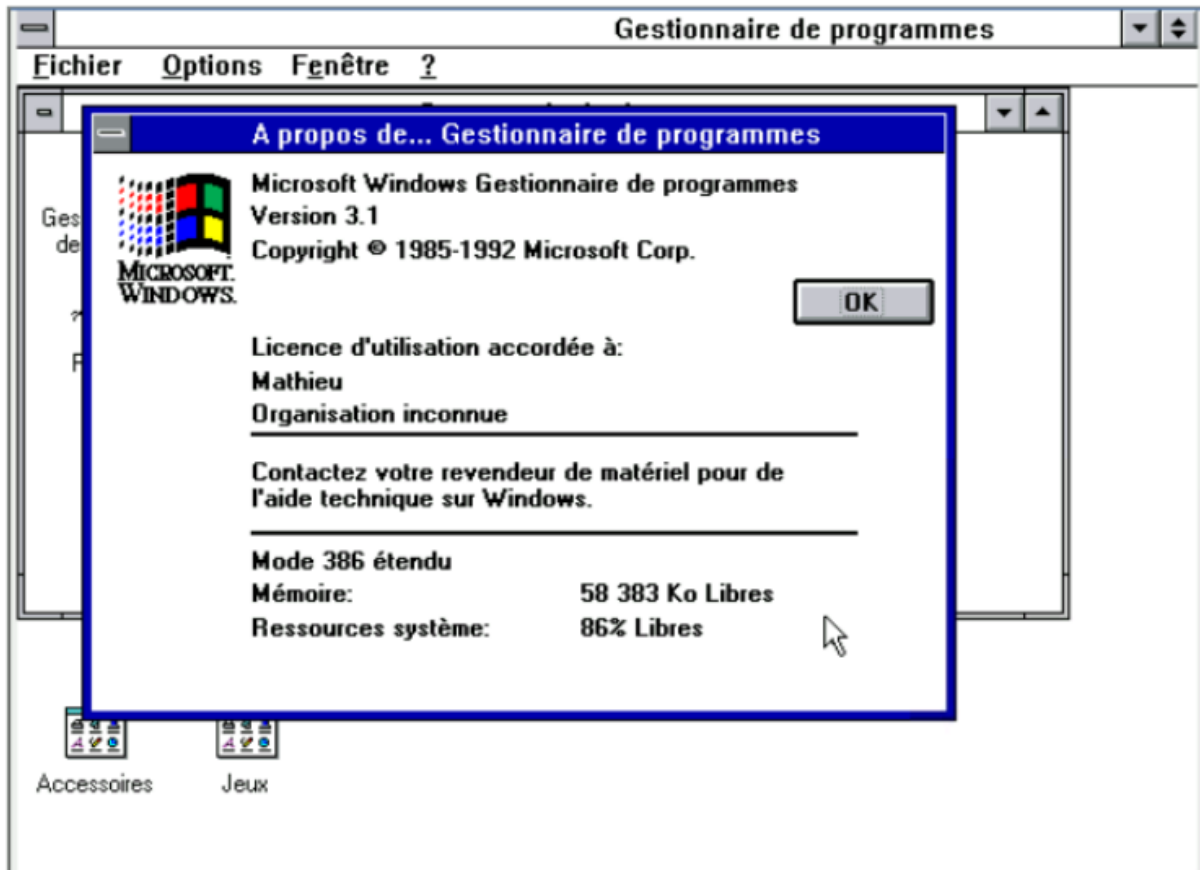


Figure N° 1.1.6: Interface de Windows 3.x

### Windows 95

Windows 95 prend des allures plus familières avec l'aspect du bureau des systèmes Windows actuels. Bonjour bureau, barre des tâches et menu démarrer.

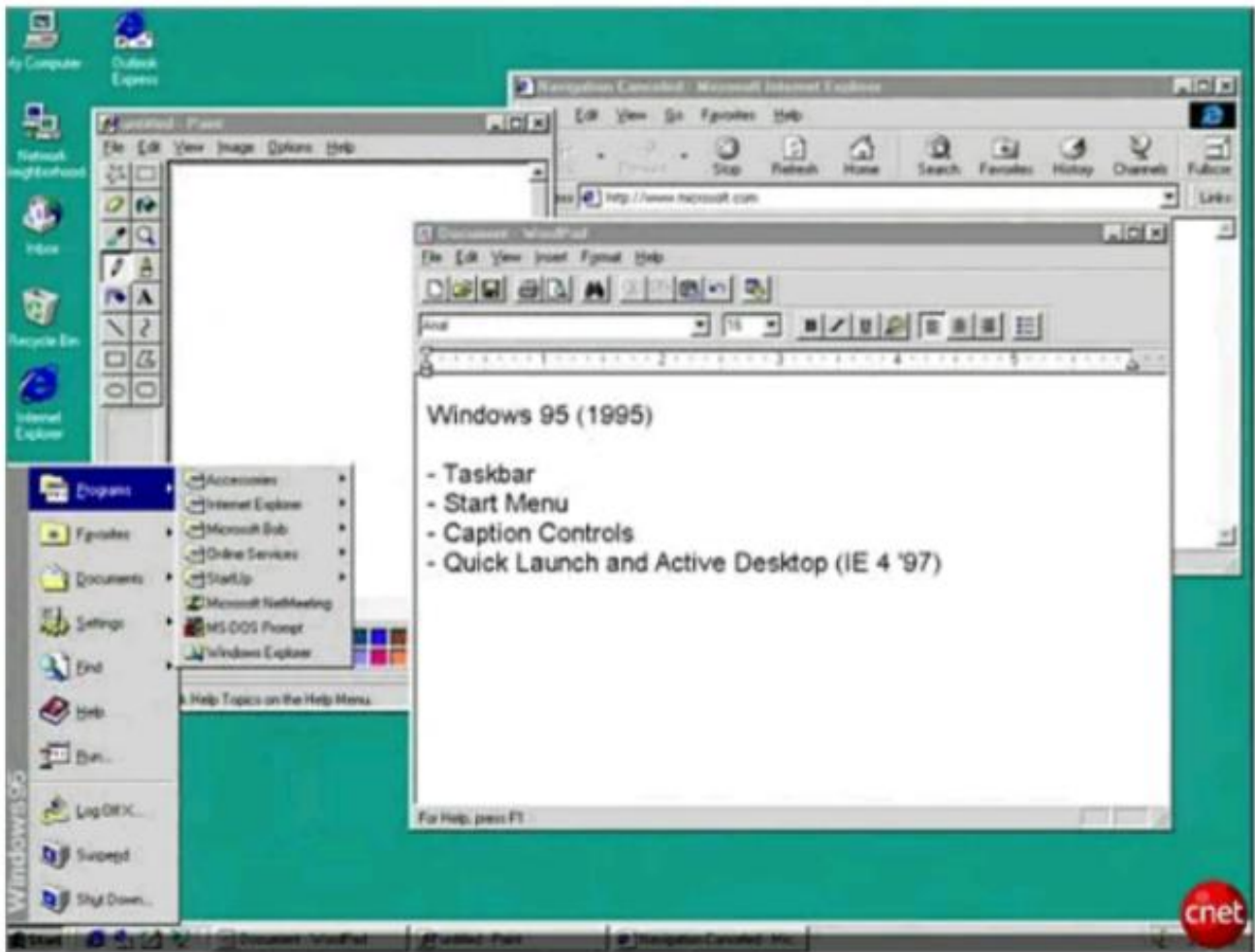
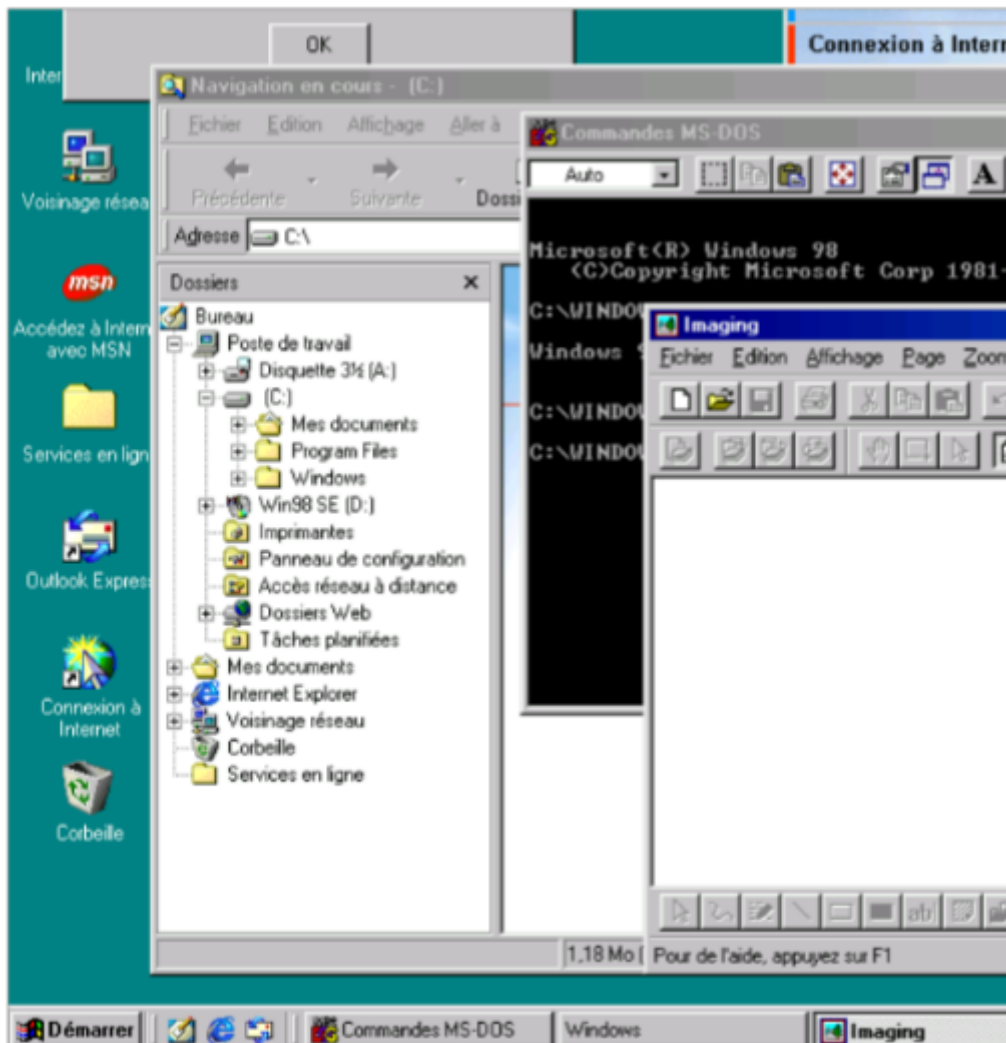


Figure N° 1.1.6: Interface de Windows 95

### Windows 98

Nous sommes toujours dans une version 4 de Windows malgré la numérotation. Le système s'ouvre à l'internet mais continue de prendre soin de son design.



*Figure N° 1.1.7: Interface de Windows 98*

### *Windows 2000/ME*

Les deux systèmes se ressemblent mais seul Windows 2000 saura séduire et s'imposer dans le temps, notamment auprès des entreprises.



*Figure N° 1.1.8: Interface de Windows 2000*

### *Windows XP*

Sorti en 2001 le système est encore utilisé de nos jours. En 2018, il est encore installé sur de nombreux ordinateurs malgré l'absence de support et donc de correctifs de sécurité. Son design est moderne et plaît au public.

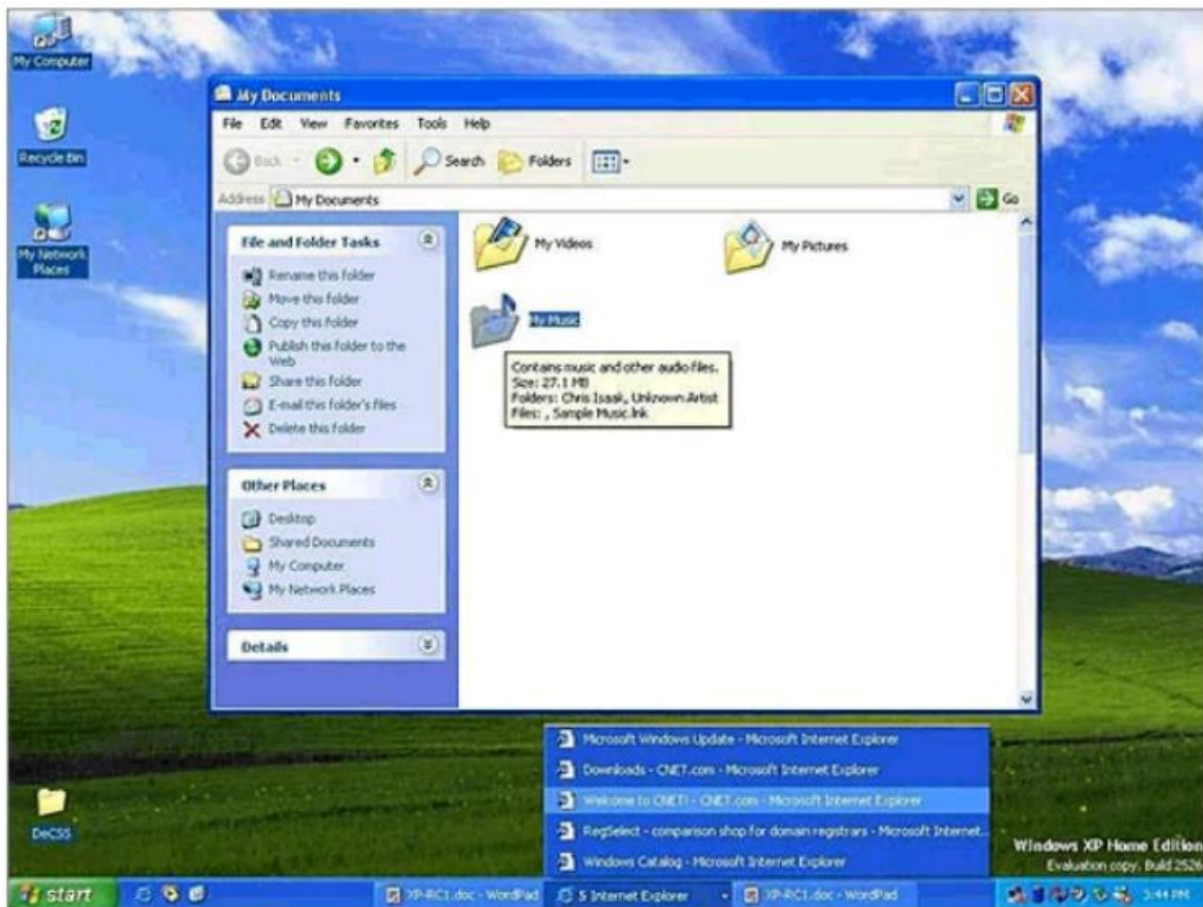


Figure N° 1.1.9: Interface de Windows XP

### Windows Vista

Un développement long (+6 ans) car remis à plat en cours de route pour cause de retard. L'interface "Aero" ne suffit pas à faire oublier la lourdeur du système et les nombreuses incompatibilités avec le matériel conçu pour Windows XP.



*Figure N° 1.1.10: Interface de Windows Vista*

### *Windows 7*

Microsoft sort Windows 7 fin 2009 avec un look de Vista moins chargé. Plus rapide et léger, il est très bien accueilli dès les premières versions. En avril 2011 il détrône Windows XP. Si c'est encore le système le plus utilisé jusqu'en 2017, Windows 10 l'a depuis dépassé.

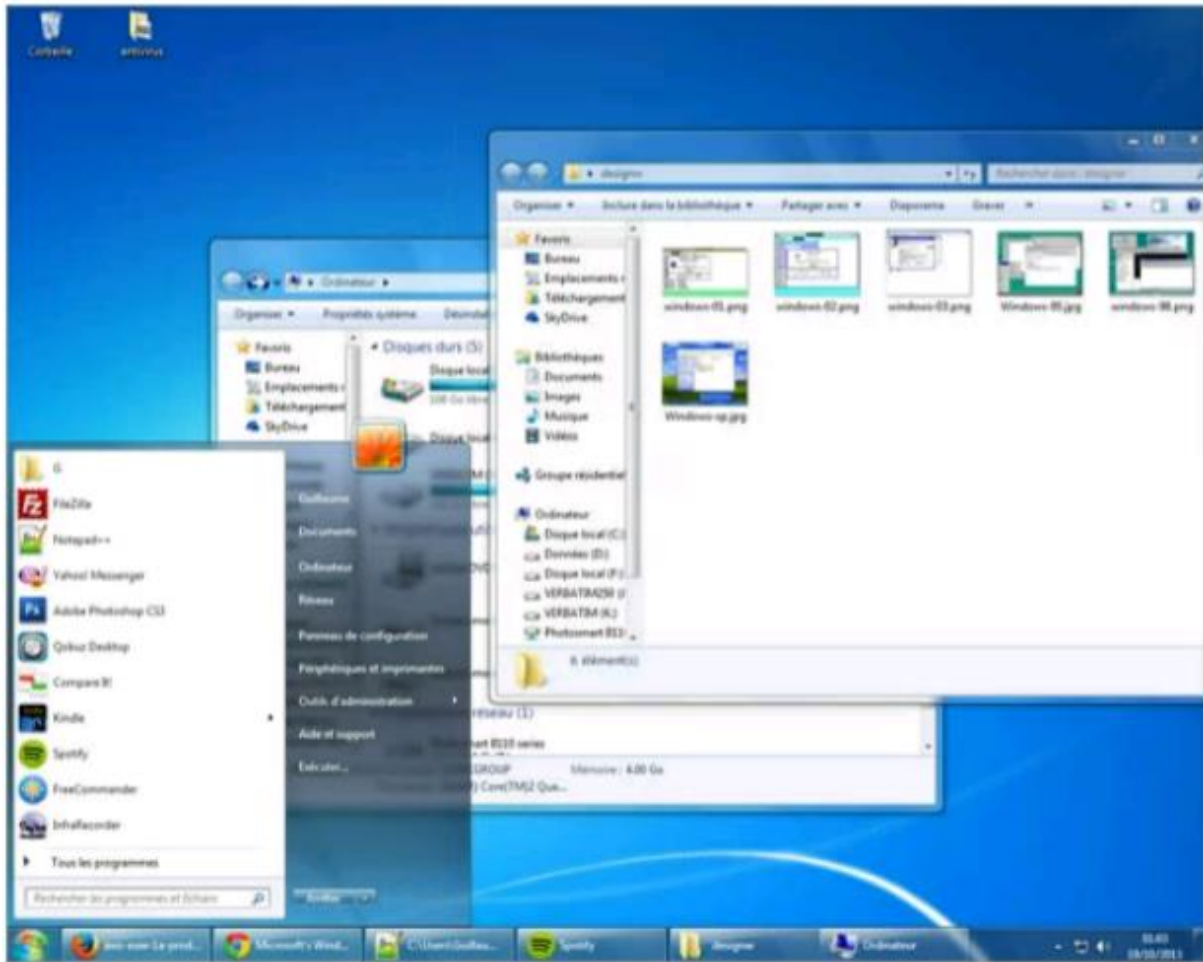


Figure N° 1.1.11: Interface de Windows 7

### Windows 8/8.1

Pour Windows 8, Microsoft ne peut plus ignorer l'essor des tablettes et autres appareils portables et dévoile un système utilisable sur les écrans tactiles. Le bouton et menu démarrer en font les frais. Un an plus tard, Windows 8.1 remet en place le bouton démarrer et ajoute la *possibilité de démarrer sur le bureau* pour éviter l'interface tactile.

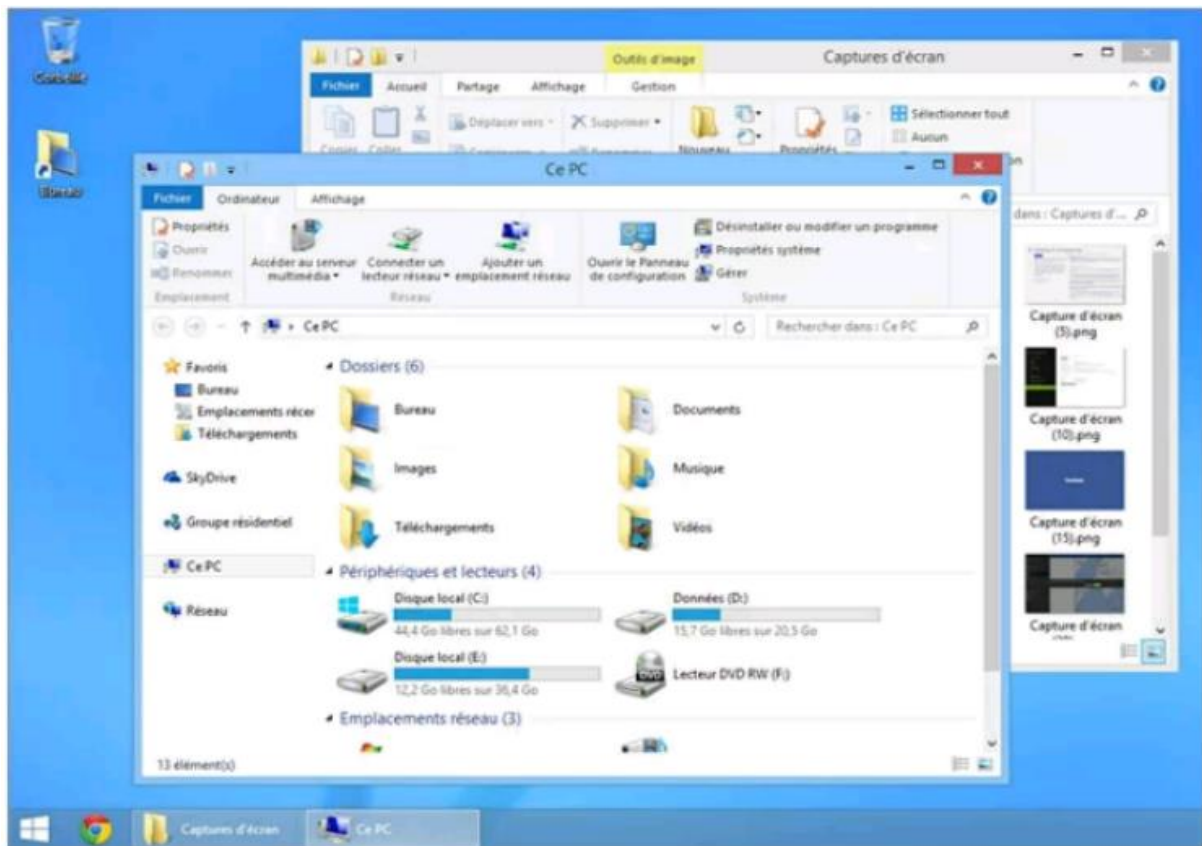
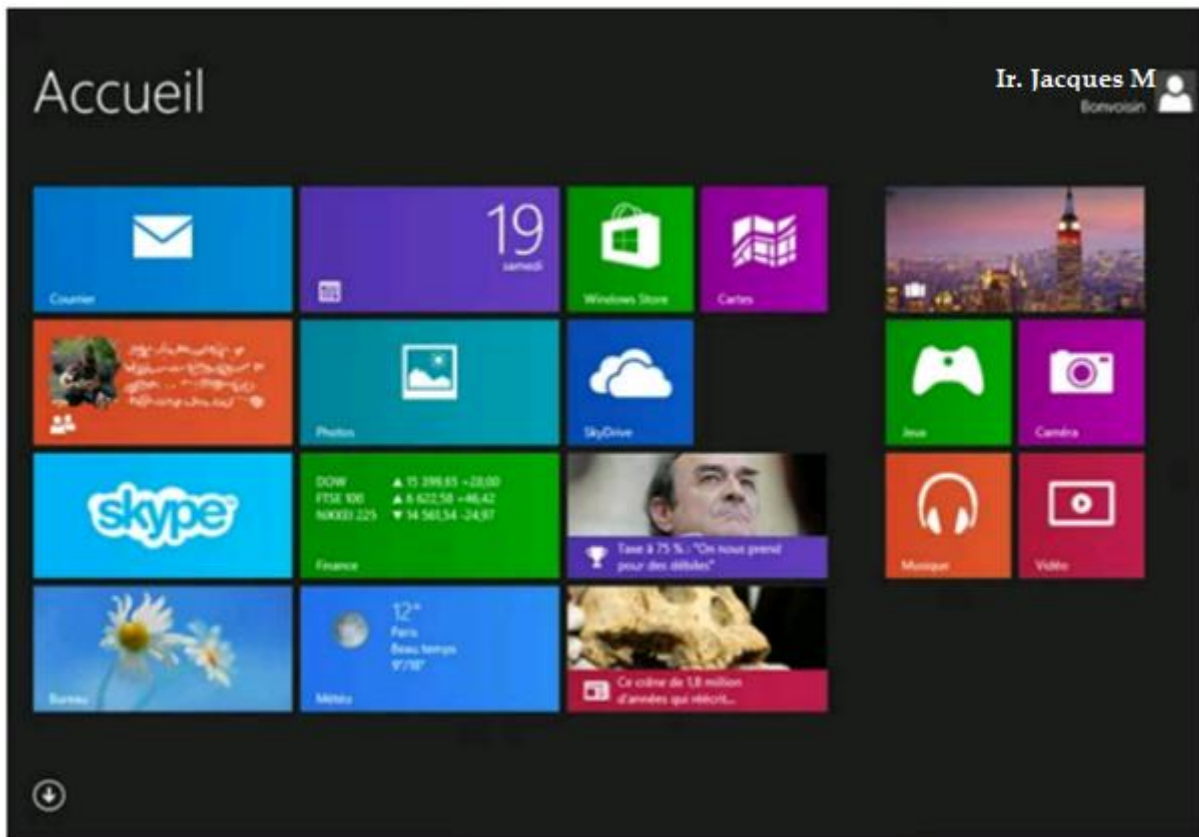


Figure N° 1.1.12 : Interface de Windows 8



## Windows 10

**Windows 10** est le dernier système en date de Microsoft qui a tenté de garder ce qui était apprécié dans Windows 7 et de conserver ce qui a plu dans Windows 8, c'est-à-dire la possibilité d'utiliser le système au tactile. Le menu démarrer est de retour, avec les tuiles, et l'utilisateur garde la possibilité de passer en mode tablette (automatiquement ou manuellement) lorsqu'il souhaite utiliser l'ordinateur avec un écran tactile et sans clavier ni souris.

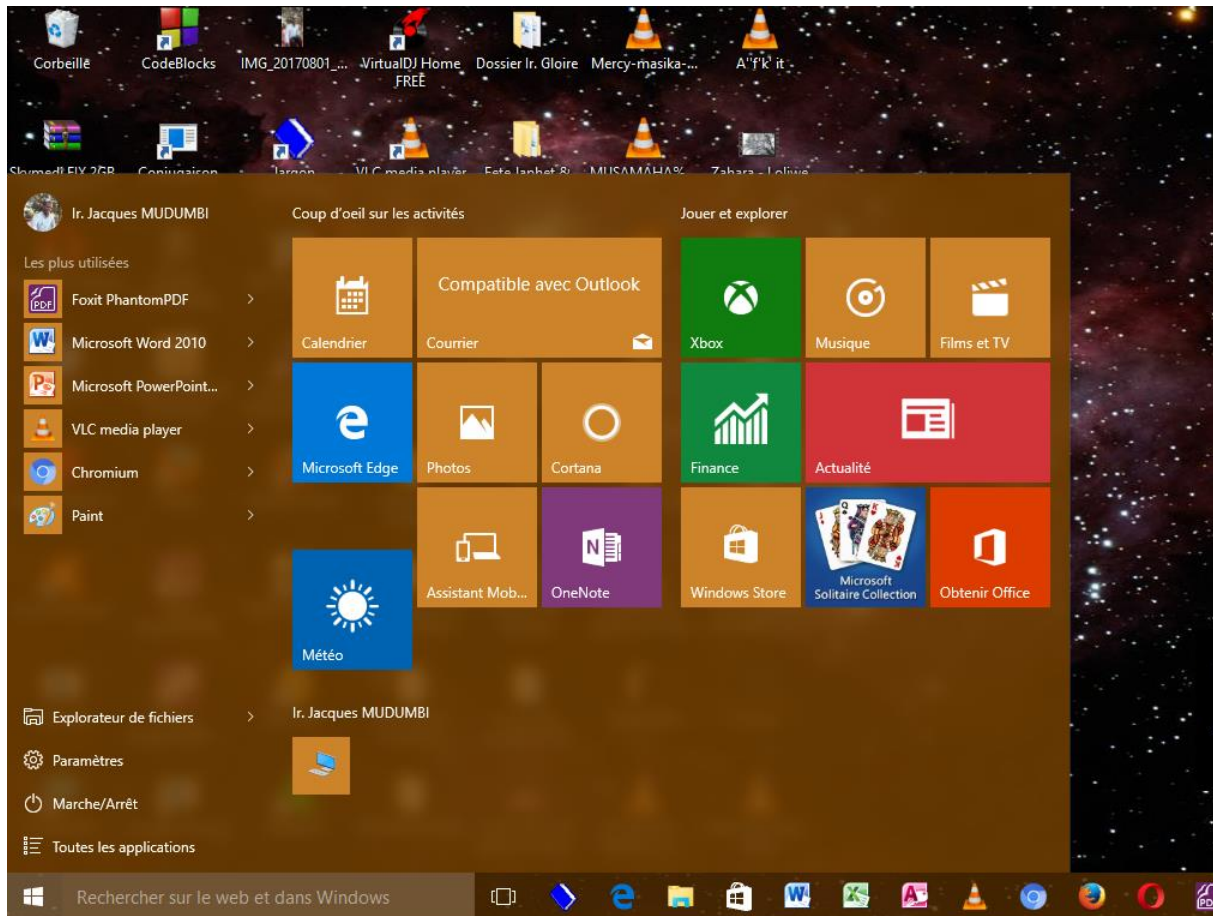


Figure N° 1.1.13 : Interface de Windows 10

## I.3.2. INTRODUCTION A UNIX

**Le système d'exploitation UNIX** est devenu un standard de l'informatique. Il se trouve au carrefour de l'informatique et des télécommunications. En effet, une de ses originalités est d'avoir su prendre en compte très tôt le phénomène des télécommunications (au travers du protocole TCP/IP). UNIX désigne une famille de systèmes d'exploitation dont le premier a été conçu aux laboratoires Bell (Bell Laboratories). C'est un système qui est assez vieux (une bonne trentaine d'années), utilisé tant pour les gros ordinateurs que pour les plus petits. Nous le retrouvons sur les super-ordinateurs (Cray), sur les ordinateurs centraux, sur les minis (VAX, HP), sur les postes de travail (HP, Apollo, Sun, SGI, ...) et bien sûr, sur les micros (Linux).

### **1960+**

Il y a beaucoup de débats sur les mérites des différents langages de programmation (PL/1, APL, Simula, ALGOL 60, COBOL, FORTRAN, etc.). À l'Université de Londres et de Cambridge, le langage BCPL (Basic Combined Programming Language) est créé. Des recherches sur les concepts de temps partagé, de traitement interactif (par opposition au traitement par lots), de pagination et de protection de la mémoire, d'ordonnancement des travaux et de structure de fichiers sont débutées.

### **1967**

Dennis Ritchie quitte Harvard pour travailler aux Laboratoires Bell dans le New Jersey.

### **1968**

Ken Thompson quitte Berkeley, où se faisaient déjà des recherches sur un nouveau système d'exploitation (SDS930), pour se joindre à une équipe de spécialistes qui avaient travaillé sur les systèmes Multics (Cambridge Multiple Access System) et GE 645.

### **1969**

Thompson et Ritchie produisent la première édition d'un système à usager unique sur un PDP 7. C'est un système primitif qui ne comporte qu'un assembleur et un chargeur.

### **1970**

La primitive « fork » est ajoutée pour permettre la création de processus et des programmes utilitaires pour la gestion des fichiers sont produits (deuxième édition). Le système accepte alors deux usagers et Thompson le baptise UNIX. Ken Kernighan avait, un jour, fait référence au système par: « Uniplexed Information and Computing System » (UNICS) par opposition à MULTICS (Multiplexed Information and Computing System).

### **1971**

Le système est transporté sur un PDP 11/20 et un système de traitement de textes (roff) est produit pour le service des brevets des Laboratoires Bell (le premier utilisateur extérieur de UNIX). Thompson et Ritchie publient la première documentation du système. C'est la troisième édition.

### **1972**

UNIX est amélioré en lui ajoutant la notion de relais ( pipe ). Thompson développe le langage B (un descendant de BCPL) et réécrit l'assembleur du système en B. C'est un compilateur qui produit du code interprétable, peu performant et Ritchie produit un générateur de code pour le PDP 11. C'est le langage C. Il existe à l'époque environ 20 sites utilisant le système UNIX.

### **1973**

UNIX est réécrit en langage C (quatrième édition). Il fonctionne alors sur un ordinateur avec un disque rigide de 500K et peut supporter environ cinq usagers. Thompson se charge de la gestion de processus tandis que Ritchie s'occupe de la gestion des entrées/sorties.

### **1974**

Plusieurs universités commencent à utiliser UNIX et la cinquième édition, conçue spécialement pour des fins académiques, est introduite.

### **1975**

La sixième édition de UNIX est produite et devient la première à être commercialisée pour une somme modique par AT&T.

### **1977**

Près de 500 sites utilisent UNIX. Une nouvelle version pour un ordinateur Interdata 8/32 est produite. Le langage C est aussi amélioré. John Reiser et Tom London, des Laboratoires Bell, écrivent UNIX 32V pour le VAX 11/780. C'est un descendant de cette version qui est actuellement distribué par l'Université de Berkeley en Californie (UCB).

À partir de ce moment, les versions de UNIX vont se multiplier; BSD4.2 de Berkeley, A/UX de Apple Computer, HP-UX de Hewlett-Packard, AIX de IBM, Domain/IX de Apollo Computer et SunOS de Sun Microsystems. C'est à cette époque que débute le problème de standard.

### **1980**

Il y a environ 100 000 sites UNIX! Le problème, c'est que personne ne supporte vraiment ce système. Les détenteurs de licences sont laissés à eux-mêmes.

### **1983**

AT&T annonce System V qui est complètement différent de la version de Berkeley.

### **1984**

La cour des États-Unis brise le monopole de AT&T et l'oblige à se subdiviser. Sept compagnies naîtront (les Baby Bells) et AT&T aura le droit de vendre des produits informatiques (de la concurrence pour IBM). Elle décide de miser sur UNIX, produit la version 2 de System V et offre un support complet pour le système d'exploitation et les utilitaires.

### **1986**

AT&T annonce la version 3 de System V qui supporte RFS (Remote File Sharing).



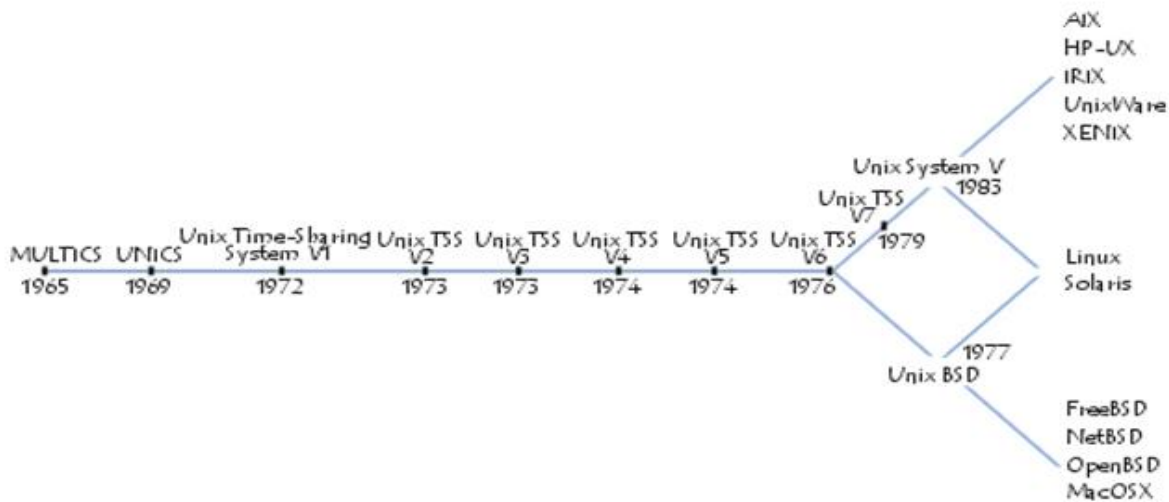


Figure N° 1.1.14 : Historique d'Unix

Le système Unics (Uniplexed Information and Computing Service) a été créé aux laboratoires AT&T de Bell, en 1969 par Ken Thompson, et modifié et baptisé par Brian Kernighan [Tan01]. Il a été une version réduite de Multics (Multiplexed Information and Computing Service). Peu après on a changé le nom Unics par Unix, et à partir de ce moment le système Unix a commencé un long chemin de développement technique. Le système Unix a connu un véritable succès, lorsqu'il fut réécrit en langage C en 1973 par Denis Ritchie et Thompson. L'Université de Californie à Berkeley a obtenu une copie de la version 6 du système Unix. AT&T et Berkeley ont séparément apporté de nombreuses modifications et améliorations au système Unix (System V d'AT&T et 4.4BSD de Bekerley).

Le projet Posix (Portable Operating System UnIX) de normalisation du système Unix a permis de développer un standard qui dénit un ensemble de procédures. Tout système Unix conforme à Posix fournit ces procédures ou appels système standards. Ces procédures constituent la bibliothèque standard d'Unix. Tout logiciel écrit en n'utilisant uniquement les procédures de la norme Posix devrait fonctionner sur tous les systèmes.

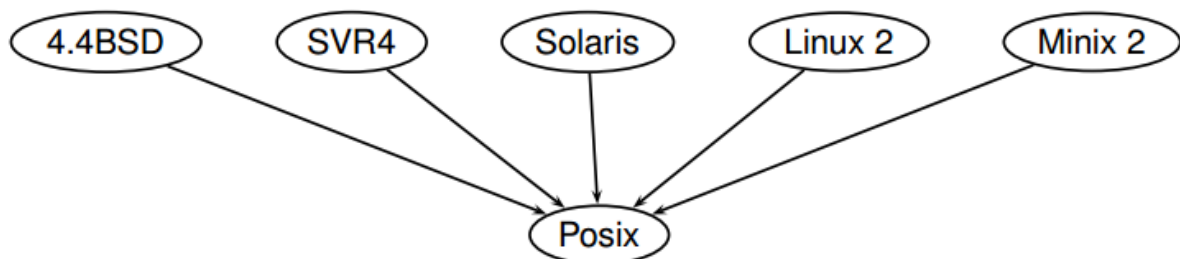
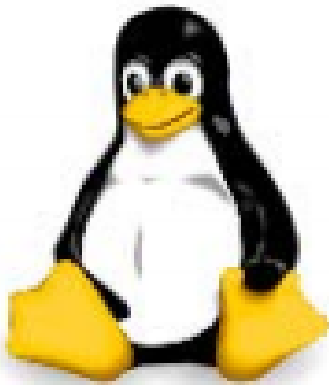


Figure N° 1.1.14 : Standard Posix

Unix est un système d'exploitation très populaire parce qu'il est présent sur un grand nombre de plates-formes, du micro-ordinateur à l'ordinateur central (mainframe). L'avantage de cela, c'est que les programmes développés sous Unix peuvent être transférés d'une plate-forme à une autre avec un minimum de modifications. Ce système est multitâche, c'est-à-dire qu'il est capable de gérer et d'exécuter plusieurs programmes simultanément. De plus, il est multi-utilisateurs, c'est-à-dire que plusieurs personnes peuvent s'y connecter en même temps et travailler ; le système partage alors toutes les ressources logicielles et matérielles de l'ordinateur entre les différents usagers. L'histoire d'Unix est unique dans le monde des systèmes d'exploitation. En effet, alors que la plupart des systèmes d'exploitation ont été conçus par des fabricants d'ordinateurs pour vendre leurs machines, Unix n'a pas été conçu dans un but commercial. Il l'est devenu parce qu'il constitue une norme en matière de système d'exploitation. Contrairement à un système d'exploitation commercial complètement contrôlé par son fabricant, le système Unix est aujourd'hui distribué par plusieurs intervenants.

## **GNU/Linux**



**GNU/Linux** est un système d'exploitation multi tâche dérivé du projet américain UNIX créé dans les années 1960 par Kenneth Thompson. Dans les années 1980, Richard Stallman, lance un projet dérivé du projet UNIX sous le nom de GNU. Cela lança le mouvement du logiciel libre basé sur le partage et l'utilisation en libre accès du code source de ces logiciels. Le fait que ces codes soient en libre accès pousse les développeurs à y apporter des modifications comme des correctifs de sécurité par exemple.

### **1991**

**Linus Torvalds, étudiant à l'université d'Helsinki (Finlande) installe le système Minix sur son i386.** Le système s'avérant trop limité pour Linus, il décide d'aller plus loin sur la base de ce qui existe... Linux (Linus' UNIX, l'UNIX de Linus) 0.0.1 est né au mois d'août 1991. Linus lance alors un appel à contribution, et permet donc un libre accès au code source.

Cette version permet de faire tourner quelques applications GNU (logiciels libres) essentielles comme le compilateur gcc ou le shell bash. Linus prend la décision de mettre le code source sous licence GPL : tout le monde peut alors participer au développement de Linux.

#### **1991-94**

Le développement anarchique de Linux ne lui permet pas, au départ, de devenir un système compétitif face aux autres systèmes du marché. Son point faible réside dans son système de fichiers hérité de Minix. Seule l'intégration à Linux du Second Extended Filesystem (ext2fs), conçu par Rémi Card à partir du système de fichiers de BSD, permet d'en faire un système fiable. Ext2fs offre enfin les performances et les services de systèmes de fichiers professionnels. Il devient naturellement un standard.

La présence dans le processus de développement de développeurs venant de nombreux horizons permet à Linux d'exister sur de nombreuses plates-formes (Mac, Atari, Amiga, Alpha) autres que sa plate-forme d'origine (Intel). Petit à petit, Linux devient un système UNIX complet compatible avec les autres systèmes UNIX, offrant toujours plus de services de qualité professionnelle au plus grand nombre.

#### **1995**

L'explosion d'Internet donne un second souffle au développement de Linux: non seulement pour permettre à la communauté des développeurs de Linux de continuer à s'étendre, mais aussi et surtout pour donner à Linux une existence réelle sur le marché des systèmes d'exploitation. Ainsi, ses qualités d'OS libre (et gratuit), robuste et performant font qu'il est choisi par de plus en plus de fournisseurs d'accès à Internet. Il est ainsi devenu aujourd'hui le leader sur le marché de l'hébergement de sites Web.

Parallèlement, l'apparition et le développement de sociétés privées telles que RedHat , Caldera ou VA Linux donne une envergure jusqu'alors inconnue à Linux: les distributions deviennent de plus en plus conviviales et simples à installer, et des services professionnels sont mis en place pour faciliter l'implantation de Linux dans les entreprises.

#### **1996**

Linux commence à faire parler de lui dans les médias. Red Hat Linux est élu meilleur OS par InfoWorld. Début du projet KDE : on commence à développer des projets conviviaux pour le grand public.

#### **1999**

Linux est présenté comme une alternative au système Windows de Microsoft dans le domaine des serveurs. Linux est présent sur 35% des serveurs d'entreprises. Les salons Linux se multiplient et l'on observe une multiplication des revues spécialisées.

### I.3.3. INTRODUCTION A MacOS



*MacOs (anciennement OS X)* a été développé par la société Apple. Ce système d'exploitation n'est présent que sur les ordinateurs de la marque Apple (Macintosh). Ceux-ci sont facilement reconnaissables grâce au logo représentant une pomme. La version actuellement vendue est MacOS Sierra, la version MacOS High Sierra devrait sortir à l'automne 2017. Outre, **Mac OS** est un système d'exploitation développé par Apple dans les années 1980 pour ses ordinateurs Macintosh. Cet OS s'adresse avant tout aux artistes ainsi qu'aux professionnels. Tout commence avec le développement du Macintosh au milieu des années 1980 par Steve Jobs et son équipe. Le Macintosh est le premier ordinateur à proposer une interface graphique en utilisant la métaphore du bureau avec des dossiers, des fichiers et une corbeille à l'écran. Dans les années 1990, Apple améliore son Macintosh pour proposer un modèle sur la base du power PC avec une tour dans laquelle sont installés les composants. En termes de systèmes d'exploitation, Apple a également modifié son système pour le rendre performant grâce à des mises à jour régulières.

Mac Os est le nom du système d'exploitation des ordinateurs de la marque à la pomme au même titre que Windows pour Microsoft. La différence notable est que Microsoft vend son système d'exploitation sans la partie matérielle alors Mac OS est toujours vendu dans une machine Apple. Le premier système d'exploitation d'Apple est né en 1984. Il ne porte pas encore le nom de « Mac OS » mais tout simplement « system 0.0 ». Entre 1984 et 1988 le « system » n'évoluera pas beaucoup et souffrira de bugs et d'une stabilité perfectible.

#### *Mac OS 8*

C'est en 1997 qu'apparaît la dénomination de « Mac OS » (Macintosh Operating System). C'est la huitième version du système d'exploitation qui hérite de ce nom. Cette version inaugure une interface graphique entièrement nouvelle (Platinum) C'est aussi cette version qui embarquera les applications pour l'arrivée d'internet.



### **Mac OS 9**

Cette version est très proche de la précédente. Elle sert de transition avant l'arrivée de Mac OS X. Elle est très aboutie et très stable et est entièrement compatible Power PC. L'interface s'est tout de même affinée.

### **Mac OS X 10.0 « Cheetah »**

C'est la première version de Mac OS X qui sort en 2001. Cette version fait apparaître le Dock et un style graphique (Aqua) fait de transparence et d'ombres qui suivront les versions ultérieures.

### **Mac OS X 10.1 « Puma »**

Il s'agit de la première grosse mise à jour de Mac OS X. Elle sort en septembre 2001. La grosse innovation réside dans sa vitesse d'exécution bien plus rapide que les versions précédentes.

### **Mac OS X 10.2 « Jaguar »**

Cette version inclut quelques nouveautés comme iChat. Graphiquement le dock perd ses rayures pour un fond uni.

### **Mac OS X 10.3 « Panther »**

Panther est encore plus rapide que son prédécesseur et adopte un style aluminium brossé pour ses fenêtres. Une colonne à gauche apparaît pour faciliter la navigation. Le changement d'utilisateur est également facilité avec un effet de cube 3D qui tourne (la classe J).

C'est également l'arrivée « d'exposé » qui permet de visualiser toutes les fenêtres ouvertes d'un seul coup d'œil.

### **Mac OS X 10.4 « Tiger »**

Tiger est toujours plus rapide et le système de recherche Spotlight fait son entrée ainsi que le dashboard (une nouvelle fonction d'exposé)

### **Mac OS X 10.5 « Léopard »**

C'est une mise à jour majeure de l'OS. L'interface graphique évolue grandement avec l'apparition des piles, d'un dock en 3D, les icônes sont entièrement revues, la présentation des fichiers sous de cover flow apparaît également.

C'est également l'arrivée de Time machine qui permet de restaurer un fichier/dossier ou le système entier de la façon dont il était quelques minutes auparavant. Une révolution !

### **Mac OS X 10.6 « Snow Leopard »**

Cette version abandonne les Power PC au profit des Intel, elle est plus stable et plus sécurisée. Les évolutions sont surtout dans le code informatique.

### **Mac OS X 10.7 « Lion »**

Pour la première fois, cet OS n'est pas vendu avec un support physique. Il faut le télécharger via l'Appstore. Ce dernier fait d'ailleurs partie des nouveautés. Le launchpad fait également son apparition. L'interface évolue également.

### **Mac OS X 10.8 « Mountain Lion »**

Parmi les évolutions, on peut noter une meilleure intégration d'icloud, l'application Messages remplace Ichat, intégration des réseaux sociaux ou encore un centre de notification.

### **Mac OS X 10.9 « Mavericks »**

C'est la première qui abandonne les appellations de félin et surtout qui est gratuite ! Une première pour Apple,

Les applications iBook et Plans (déjà disponibles sur iOS) font leur apparition, l'interface de Safari est renouvelée

### **Mac OS X 10.10 « Yosemite »**

Dans cette version l'interface graphique évolue significativement pour suivre la tendance du flat design. Les fenêtres sont plus claires, un mode nuit apparaît et on note l'ajout de flou et de transparence.

Spotlight, Safari font peau neuve. Il est possible d'envoyer des SMS/Appels depuis le Mac grâce à un iPhone synchronisé.

### **Mac OS X 10.11 « El Capitan »**

Cette nouvelle version se concentre essentiellement sur l'ergonomie et les performances. On note parmi les nouveautés :

Split view pour travailler sur 2 fenêtres d'application en même temps.

Le curseur d'attente a été revu, Safari peut muter le son d'un onglet ;

### **Mac OS X 10.12 « Sierra »**

Sierra inaugure un nouveau système de fichiers 64bits. Ce dernier est plus adapté aux SSD. On note également l'arrivée de Siri et d'Apple Pay.

### **Mac OS X 10.13 « High Sierra »**

C'est la version actuelle de Mac OS X,

Pour la vidéo, Mac OS prend en charge le codec H265 (4K). On note quelques nouveautés mineures par rapport à Sierra.

## En bref que peut-on retenir de ces trois principaux systèmes d'exploitation ?

Chaque système d'exploitation a sa propre présentation et son ergonomie, mais surtout, cela a un impact sur l'utilisation des logiciels. En effet, certains logiciels ne peuvent être utilisés que sur MacOS ou Windows. N'hésitez pas à vous renseigner si vous voulez acheter un ordinateur.

**Microsoft Windows**, le système d'exploitation le plus utilise **Windows**, le système d'exploitation développé par **Microsoft**, est le plus utilisé au monde et il est souvent livré avec l'ordinateur. Facile d'accès et convivial, il est décliné en différentes versions (on en est actuellement à Windows 8) et en différentes versions (grand public, professionnel, entreprise). Le nombre de logiciels et de jeux développés pour Windows est phénoménal, et c'est l'OS le plus adapté pour une utilisation grand public. Avec un ordinateur Windows, vous n'aurez aucun mal à utiliser les logiciels du marché, et vous pourrez le faire fonctionner sur de nombreux appareils différents. À l'usage, Windows est souvent la cible d'attaques virales, du fait de sa popularité, et il faut fréquemment le mettre à jour. Ce système d'exploitation est vendu sur différentes marques d'ordinateurs (**Acer, Asus, Dell, HP, Sony, Toshiba...**)

## **Linux, le système d'exploitation Open Source**

Le système d'exploitation Linux est réservé aux utilisateurs qui utilisaient un logiciel libre plutôt que les deux OS leaders du marché. Il existe plusieurs centaines de distributions Linux à télécharger gratuitement sur Internet, mais il est conseillé de débiter par Ubuntu si on a des connaissances informatiques de base. L'installation de Linux est simple et conviviale sur un ordinateur PC ou Mac, et il existe de nombreux tutoriels sur Internet. Attention, si vous trouverez facilement des équivalents de tous les logiciels de bureautique, il n'est pas toujours simple de trouver une alternative Linux à un logiciel développé pour Windows, sans parler des jeux vidéo ! Par exemple, il n'y a pas de réel équivalent à Adobe Photoshop et Illustrator, alors évaluez bien vos besoins avant de vous lancer !

## **Mac OS, le système d'exploitation proposé par Apple**

Si vous êtes intéressé par le design et l'ergonomie, et que vous utilisez surtout votre ordinateur à des fins professionnelles, vous pouvez vous laisser tenter par Mac OS, un système d'exploitation payant qui est la propriété d'Apple, et qui équipe tous les ordinateurs de la marque. La disposition du menu et la gestion des programmes sont un peu différentes de celle de Windows, mais les interfaces graphiques tendent de plus en plus à se ressembler. Tout comme Microsoft, Apple propose régulièrement de nouvelles versions de son OS (la dernière version est Mac OS 10.10 dit "Yosemite"). Sachez-le aussi, si vous possédez un ordinateur Mac, il est possible d'installer un autre système d'exploitation dessus !

Le système Mac OS s'est de plus en plus démocratisé mais n'atteint pas le même taux d'adoption que celui de Windows. Cela est dû à la politique de prix élevée chez Apple dans la

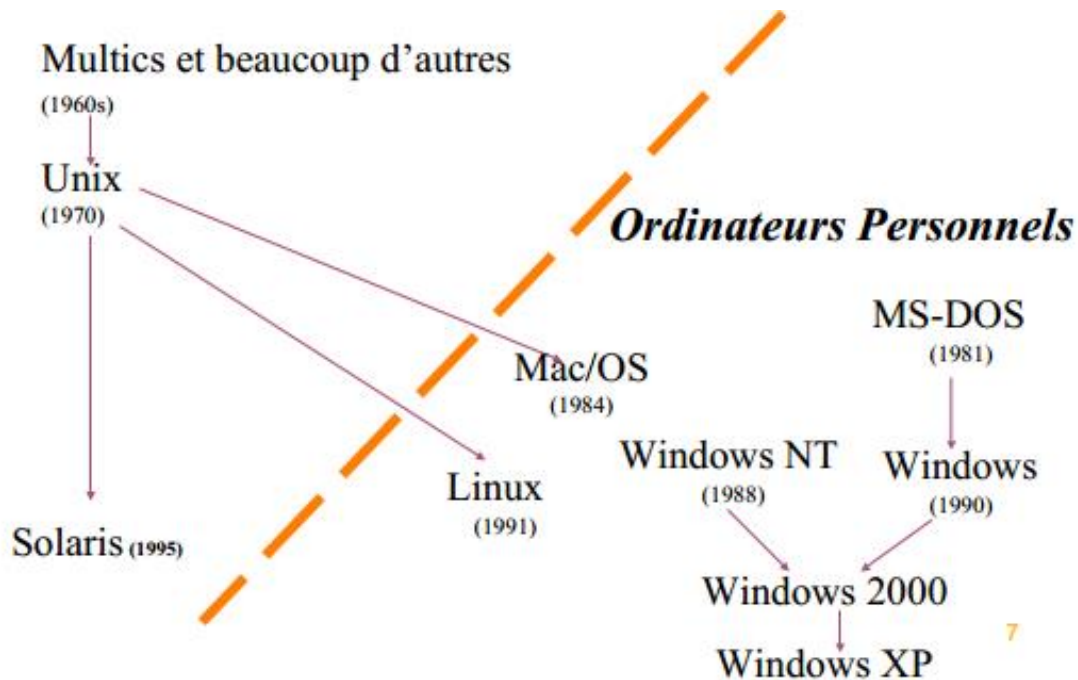


Figure N° 1.1.15 : Généralité sur les versions des principaux systèmes d'exploitation

L'étude de ces 3 systèmes d'exploitation **Windows, Linux et MacOs** nous aura permis d'appréhender les fonctionnalités et les caractéristiques de ces trois systèmes d'exploitation. Cependant, il appartient à l'utilisateur de choisir en fonction des objectifs et besoins souhaiter l'utilisation du système d'exploitation voulu. **Outre, Windows, macOS et Linux**, ces systèmes d'exploitation ont évolué avec le temps, il en existe donc plusieurs versions. Comparons maintenant les principaux systèmes entre eux ;

En général, les systèmes d'exploitation sont préinstallés et prêt à l'emploi. Cela est surtout valable dans le cas de Windows et de Mac. Ces deux systèmes représentent respectivement 90 % du marché pour Windows et 8% pour Mac. GNU/Linux ne possède que 2% du marché et les ordinateurs équipés d'une des distributions sont rares. Généralement, la distribution la plus commune est Ubuntu de chez Canonical.

Voici un tableau comparatif qui vous présente les avantages, inconvénients et nombre de bits des instructions des programmes qui sont développés pour fonctionner avec tel ou tel système ainsi que leurs caractéristiques d'utilisation matérielles principales:

## I.4. LE PARALLELISME DES SYSTÈMES D'EXPLOITATION (WINDOWS, LINUX ET MAC OS) SE DESSINE À TRAVERS LE TABLEAU COMPARATIF CI-DESSOUS

Système d'exploitation	Avantages	Inconvénients	Codage et Utilisation
Windows	<ul style="list-style-type: none"> <li>- Windows est le système d'exploitation le plus répandu.</li> <li>- Fonctionne un peu mieux que les précédentes versions.</li> <li>- On trouve des PCs à des prix très bas, mais attention à bas prix</li> <li>- Idéal pour les jeux high-tech en 3D (à condition d'avoir une machine et une carte graphique très puissantes)</li> <li>- Reconnaît tous les matériels exotiques bon marché.</li> <li>- L'entretien des PC sous Windows fournit du travail au secteur de la maintenance informatique.</li> </ul>	<ul style="list-style-type: none"> <li>- Peu des logiciels intégrés,</li> <li>- Pas de récupération des paramètres systèmes des utilisateurs, tels que les mots de passe réseau par exemple.</li> <li>- Installation obligatoire d'un anti-virus.</li> <li>- Manque de réactivité globale comparé aux Unix.</li> </ul>	<p>32 et 64 Bits</p> <p>Utilisation sur des ordinateurs personnels et stations de travail, tablettes et assistants personnels et serveurs.</p>
Linux	<ul style="list-style-type: none"> <li>- Les logiciels de Linux sont presque gratuits.</li> <li>- S'installe sur tous PC et mac en complément ou remplacement de windows.</li> <li>- Possibilité d'essais sans installation via un Live CD,</li> <li>- Puissant (car léger et très optimisé)</li> <li>- fiable et sécurisé, système libre et open source (système unix)</li> <li>- firewall intégré</li> <li>- Pas de virus</li> </ul>	<ul style="list-style-type: none"> <li>- Les matériels très récents ne sont supportés qu'après plusieurs mois.</li> <li>- Peu de jeux.</li> <li>- Certains matériels exotiques nécessitent quelques recherches web pour être paramétrés.</li> </ul>	<p>32 &amp; 64 bits</p> <p>Utilisation sur tous.</p>

	<ul style="list-style-type: none"> <li>- Compatible avec le « monde » windows.</li> <li>- Personnalisable à volonté.</li> <li>- Pas besoin d'entretien, le système peut tourner 24h/24 pendant des années sans s'engorger.</li> <li>- Système libre Open Source de type Unix.</li> <li>- Respect de votre vie privée et sécurité pour vos données.</li> </ul>		
<p>Mac Os X ou macOS</p>	<ul style="list-style-type: none"> <li>- Un système puissant, fiable et sécurisé, un firewall discret et efficace qui se fait oublier c'est-à-dire complet.</li> <li>- Pas de virus, pas de spywares si vous n'installez pas de logiciels issus de sites douteux.</li> <li>- Écrans d'excellente qualité avec des angles de vision très larges.</li> <li>- Suite logicielle intégrée (Ilife) très complète et intuitive (itunes, iphotos, imovies, etc.) permet de gérer musique, photos, vidéos et de créer facilement site web et blog.</li> <li>- Reconnaît les périphériques les plus récents (appareil photo etc.)</li> <li>- Mac OS-X basé sur un noyau BSD offre la fiabilité et la puissance d'unix et une grande diversité de logiciels.</li> <li>- Sauvegardes automatiques de tout le Mac, système, programmes, données avec Time Machine.</li> <li>- Respect de votre vie privée et sécurité pour vos données, possibilité de Crypter tout le Mac.</li> </ul>	<ul style="list-style-type: none"> <li>- Mac Os X est indissociable de la plateforme Mac : il faut donc acheter un Mac pour en bénéficier, à partir de 499€ sans écran ni clavier ni souris pour un mini mac de base incluant le graveur de CD</li> <li>- Un peu plus cher qu'un PC, car il n'existe pas de machine bas de gamme chez Apple.</li> <li>- Garantie étendue AppleCare Obligatoire si vous voulez éviter les ennuis. (Intéressant à partir de l'Imac et du Macbook).</li> <li>- Catalogue de jeux limités.</li> </ul>	<p>32 bits</p> <p>Utilisation sur des ordinateurs personnels, serveurs, station de travail, téléphones, tablettes (ios).</p>

	<ul style="list-style-type: none"> <li>- Pas besoin d'entretien, le système peut tourner 24h/24 pendant des années sans s'engorger.</li> <li>- Design sobre et excellente finition font que les Mac traversent le temps sans paraître obsolètes.</li> <li>- Conception hardware silencieuse (sauf cartes 3D très puissantes en action).</li> <li>- Compatible avec le monde Windows.</li> <li>- Possibilité d'installer Windows (l'architecture interne est la même que celle d'un PC).</li> </ul>		
--	--	--	--

## I.5. PREDILECTION DE L'OS PAR L'UTILISATEUR

Lorsque l'on achète un nouvel ordinateur, le choix du système d'exploitation n'est pas toujours très simple. Outre le coût, chaque système d'exploitation présente des avantages et des inconvénients, qui se révéleront lors de la pratique au quotidien. Le système d'exploitation ou OS (Operating System) est un ensemble de logiciels permettant de piloter l'ordinateur et ses périphériques, et d'agir sur les fichiers. Autant dire que sans OS, vous ne pourrez pas faire grand-chose ! Il existe plusieurs choix possibles, en fonction de ses besoins et de son niveau informatique.

L'utilisateur a un grand choix de système d'exploitation. Ceux-ci sont plus ou moins faciles d'accès dans leur utilisation et plus ou moins personnalisables. Tout dépend de ses goûts en matière d'informatique. Là où la plupart y verront un système prêt à l'emploi et sans prise de tête, d'autres y verront une liberté réduite dans la customisation de sa machine.

Si vous, oui vous, utilisateur, voulez un système sans prise de tête et prêt à l'emploi, optez pour un PC avec Windows ou un Mac de chez Apple. En revanche, si vous voulez personnaliser entièrement votre système, optez plutôt pour une machine avec une des nombreuses distributions GNU/Linux. Vous êtes servi quitte à y passer un peu de temps à personnaliser, vous y trouverez votre bonheur. L'installation n'est pas bien compliquée, vous avez une communauté prête à vous aider quel que soit votre souci. **Bref, Le système d'exploitation le mieux à utiliser dépend de ce que son utilisation sera destinée et qui vont l'utiliser.**

## I.6. EVOLUTION DES SYSTEMES D'EXPLOITATION

### 1. Première génération (1945-55) : les tubes à vide les cartes enfichables,

Il n'existait pas de système d'exploitation. Les utilisateurs travaillaient chacun leur tour sur l'ordinateur qui remplissait une salle entière. Ils étaient d'une très grande lenteur. Ils étaient d'une très grande fragilité. La première génération est le début du développement. Les systèmes informatiques électroniques en tant que substitut des systèmes informatiques mécaniques, c'est en raison de la vitesse de calcul des humains qui est limitée et les humains sont très faciles à faire preuve de négligence, d'erreurs et même d'erreurs. Dans cette génération, il n'y a pas de système d'exploitation, le système informatique reçoit des instructions qui doivent être effectuées directement.

### 2. Deuxième génération (1955-65) : les transistors et le traitement par lots,

Le passage aux transistors rendait les ordinateurs plus fiables. Ils pouvaient être vendus à des utilisateurs (grandes compagnies, université ou administrations). Mais devant les coûts d'équipement élevés on réduisit les temps grâce au traitement par lots. La deuxième génération introduit le traitement par lots Système, qui est un travail qui est effectué en série, puis exécuté séquentiellement. Dans cette génération, le système informatique n'a pas été équipé d'un système d'exploitation, mais certaines fonctions du système d'exploitation existent déjà, par exemple les fonctions du système d'exploitation sont FMS (Fortran Monitor System) et IBSYS.

### 3. Troisième génération (1965-80) : les circuits intégrés et la multiprogrammation,

Amélioration des coûts et des performances (circuits intégrés). Une famille d'ordinateurs compatibles entre eux. Une seule architecture et un même jeu d'instructions. Des ordinateurs uniques pour les calculs scientifiques et commerciaux. Apparition du spoule (spool, Simultaneous Peripheral Operation On Line) pour le transfert des travaux des cartes vers le disque. Apparition de la multiprogrammation (partitionnement de la mémoire pour des tâches différentes).

Dans cette génération, le développement du système d'exploitation développé pour servir de nombreux utilisateurs à la fois, où les utilisateurs interactifs communiquent via le terminal en ligne avec l'ordinateur, le système d'exploitation devient multi-utilisateur (utilisé par de nombreux utilisateurs à la fois) et multi-programmation (desservant plusieurs programmes à la fois).

### 4. Quatrième génération (1980-aujourd'hui) : les micro-ordinateurs ;

Ils sont dus au développement des circuits LSI (Large Scale Integration) contenant des centaines de transistors au cm<sup>2</sup>. Ils ont la même architecture que les mini-ordinateurs mais leur prix est beaucoup moins élevé. Il existe deux systèmes d'exploitation principaux : MS-DOS (Microsoft Inc.) et UNIX.



Apparition des PIC (Personal Intelligent Communicator de chez Sony) et des PDA (Personal Digital Assistant, comme le Newton de chez Apple), grâce à l'intégration des composants et l'arrivée des systèmes d'exploitation de type « micro-noyau ». Ils sont utiles pour les « nomades » et les systèmes de gestion des informations (recherche, navigation, communication). Ils utilisent la reconnaissance de caractère (OCR) et les modes de communication synchrone et asynchrone (mode messagerie).

Très bon marché, ils sont capables de se connecter à des ordinateurs distants et performants. Les systèmes d'exploitation de type « micro-noyau » sont modulaires (un module par fonction) ; ils peuvent être réalisés avec plus ou moins de modules et donc adaptables à des très petites machines (PDA et PIC).

Aujourd'hui, le système d'exploitation est utilisé pour réseau informatique où l'utilisateur est au courant de l'existence d'ordinateurs connectés les uns aux autres. A cette époque, les utilisateurs ont également été sécurisés avec une interface utilisateur graphique qui est une interface informatique graphique très confortable, à cette époque a également commencé l'ère de l'informatique distribuée où les calculs ne sont plus centrés sur un point, mais sont décomposés sur de nombreux ordinateurs afin que les performances soient atteintes le meilleur.

## CHAPITRE II : STRUCTURES ET ARCHITECTURES DES SYSTEMES D'EXPLOITATION

Le système d'exploitation se situe entre le matériel et les autres logiciels. Afin de rendre le travail administratif possible, le système d'exploitation a une structure légèrement différente de celle de la plupart des autres programmes. On dit que les systèmes sont construits à partir de différentes couches. Dans la couche inférieure - celle qui est la plus éloignée de l'interface utilisateur - se trouve le **noyau**, l'élément le plus important du système d'exploitation. Ce programme est donc aussi le premier à être chargé. Le noyau est l'interface directe avec le matériel, l'initialise et transmet les commandes des programmes en cours d'exécution au matériel. Ce **noyau** est ensuite utilisé comme base pour les autres couches, qui s'éloignent de plus en plus de l'interaction avec le matériel. Chaque couche communique uniquement avec la couche située au-dessus ou au-dessous d'elle. Enfin, au sommet se trouve **l'interface utilisateur**, l'interface entre les utilisateurs et le logiciel. Si l'utilisateur effectue une action, cette instruction est guidée à travers les différentes couches jusqu'à ce qu'elle atteigne le bon endroit, le processeur, par exemple.

Le système d'exploitation est le composant logiciel fondamental d'un ordinateur, **il fonctionne en mode noyau dit aussi mode superviseur**. Dans ce mode, il a un accès complet et total à toutes les ressources matérielles et peut exécuter n'importe quelle instruction que la machine est capable de traiter. Tous les autres **logiciels fonctionnent en mode utilisateur**, dans lequel une partie seulement des instructions machines sont accessibles. En particulier les instructions qui affectent la commande de la machine ou les entrées/sorties sont inaccessibles en mode utilisateur.

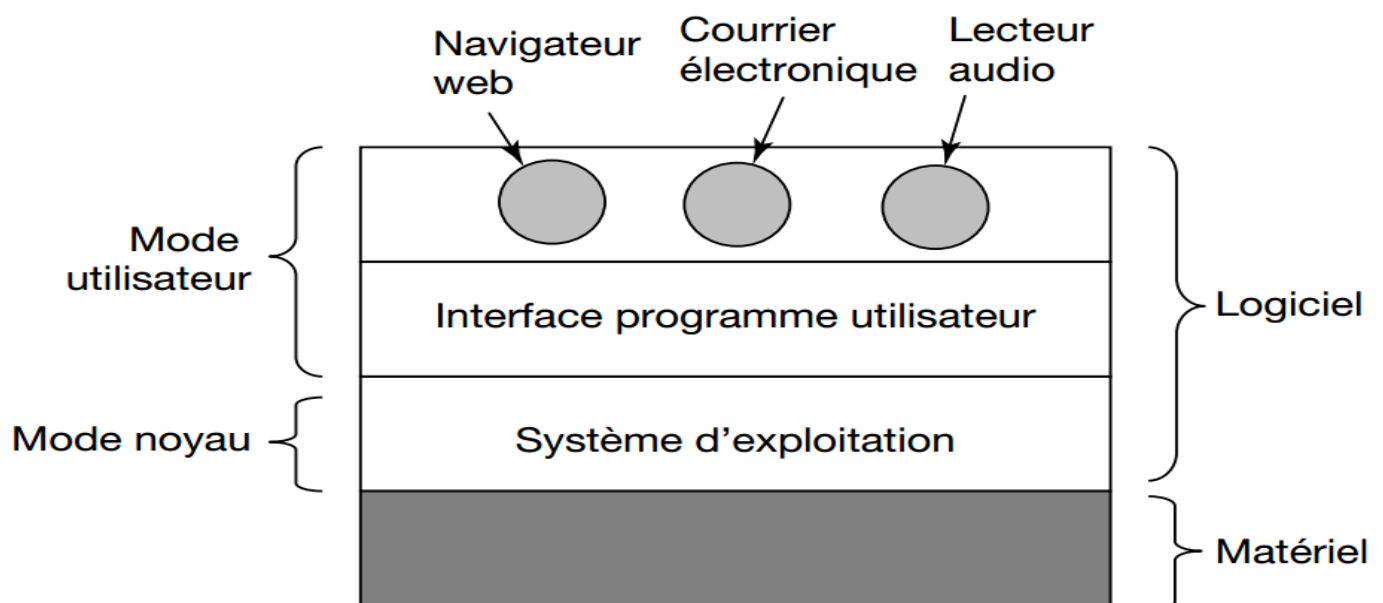


Figure N° 2.1.0 : Place du système d'exploitation au sein d'un ordinateur

## II.1. COMPOSANTS DU SYSTEME D'EXPLOITATION

Il regroupe un ensemble d'éléments que intervient dans la gestion des interactions avec le hardware. D'où, il existe les notions tels que :

- ☑ **Le noyau (kernel)** représentant les fonctions fondamentales du système d'exploitation telles que la gestion de la mémoire, des processus, des fichiers, des entrées-sorties principales, et des fonctionnalités de communication.
- ☑ **L'interpréteur de commande (shell - coquille** par opposition au noyau) permettant la communication avec le système d'exploitation par l'intermédiaire d'un langage de commandes, afin de permettre à l'utilisateur de piloter les périphériques en ignorant tout des caractéristiques du matériel qu'il utilise, de la gestion des adresses physiques, etc.
- ☑ **Le système de gestion de fichiers (file system)** permet d'enregistrer les fichiers dans une arborescence.
- ☑ **L'interface-Homme-Machine (IHM) :** qui permet de définir la manière dont un composant informatique peut communiquer avec un autre.
- ☑ **Les pilotes :** qui aident à la prise en charge des périphériques, ou c'est un ensemble de programmes qui permettent le fonctionnement du système : services réseau, accès aux fichiers,...

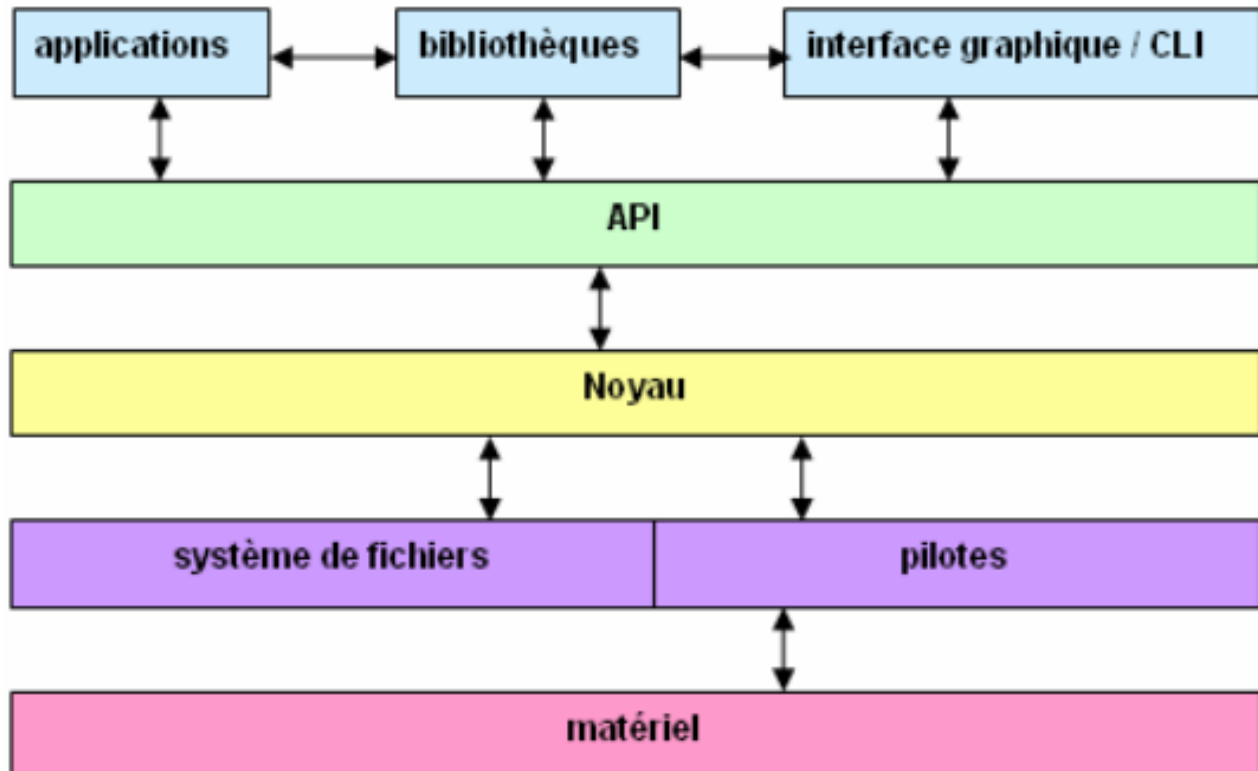


Figure N° 2.1.1: Composants d'un système d'exploitation

## II.2. LES TYPES DE SYSTEME D'EXPLOITATION

On distingue différents types: selon qu'ils sont capables de gérer simultanément des informations d'une longueur *de 16 bits, 32 bits, 64 bits ou plus* ; Il peut s'agir de faire une catégorisation, selon les services rendus, leur architecture matérielle qui les supporte et selon le cas particulier.

### 1) SELON LE SERVICE RENDUS

#### Systèmes mono/multitâches/préemptifs

- ☑ **Mono-tâche** : ce système n'autorise l'allocation du processeur que pour une seule tâche. En effet, une tâche ne prend le processeur que si seulement ou si la précédente est achevée, si non le système se bloque.
- ☑ **Multi-tâches**: capacité du système à pouvoir exécuter plusieurs tâches « processus » (i.e. « un programme ») simultanément ou un utilisateur lance plusieurs applications simultanément. *e.g.* effectuer une compilation et consulter le fichier source du programme correspondant. *Cas d'UNIX et Windows 95.*
- ☑ **Les systèmes d'exploitation préemptifs** : gèrent le temps processeur alloué à chaque application. Un commutateur de tâches intervient pour répartir l'allocation des ressources. Des degrés de priorité sont accordés à chaque application. Chaque application peut être interrompue sans interférer avec les autres applications.

#### Systèmes mono/multi-utilisateurs :

- ☑ **Mono-utilisateur** : le système ne peut gérer qu'un seul utilisateur (*e.g. le système MS-DOS*) ;
- ☑ **Multi-utilisateurs**: capacité à pouvoir gérer un panel d'utilisateurs utilisant simultanément les mêmes ressources matérielles. *Outre, plusieurs utilisateurs utilisent le même ordinateur en même temps.*

### 2) SELON L'ARCHITECTURE MATERIELLE QUI LES SUPPORTE

**Architecture monoprocesseur (temps partagé ou multi-programmation)** : Ressource processeur unique : Il a fallu développer un mécanisme de gestion des processus pour offrir un (pseudo) parallélisme à l'utilisateur : c'est la multi-programmation ; il s'agit en fait d'une commutation rapide entre les différents processus pour donner l'illusion d'un parallélisme. Ce sont des systèmes qui garantissent le partage équitable du temps processeur et des ressources dans le but *de maximiser le temps de traitement et de réduire le temps de réponse moyen.*

**Systèmes multi-processeurs** : Le multiprocessing est une technique consistant à faire fonctionner plusieurs processeurs en parallèle afin d'obtenir une puissance de calcul plus importante que celle obtenue avec un processeur haut de gamme ou bien afin d'augmenter la disponibilité du système (*en cas de panne d'un processeur*).

### 3) CAS PARTICULIER : SYSTEMES EMBARQUES, SYSTEMES TEMPS REEL

- ☑ **Systèmes embarqués** : sont des systèmes d'exploitation prévus pour fonctionner sur des machines de petite taille, telles que des PDA (personal digital assistants ou en français assistants numériques personnels) ou des appareils électroniques autonomes (sondes spatiales, robot, ordinateur de bord de véhicule, etc.), possédant une autonomie réduite. Ainsi, une caractéristique essentielle des systèmes embarqués est leur gestion avancée de l'énergie et leur capacité à fonctionner avec des ressources limitées. *e.g* : **BlackBerry OS; Android; Symbian; Lumia; Java; iOS.**
- ☑ **Systèmes temps réel** : ce sont des systèmes pour lesquels l'exécution des programmes est soumise à des contraintes temporelles. Les résultats de l'exécution d'un programme n'est plus valide au-delà d'un certain temps connu et déterminé à l'avance. Généralement on trouve ces systèmes TR dans les systèmes embarqués. *e.g* : **RTLinux (RealTime Linux), QNX ; VxWorks.**

## II.3. CLASSES DES SYSTEMES D'EXPLOITATION

Dans la plupart des systèmes d'exploitation, les critères sont considérés comme élément fédérateur de catégorisation des classes, cette catégorisation des classes peut être du soit à, aux:

- ☑ **Les interfaces:**
  - ∅ La ligne de commande, le mode texte avec le clavier.
  - ∅ L'interface graphique (GUI pour Graphical User Interface), le mode graphique, avec un pointeur comme une souris.
- ☑ **Le nombre d'application qui tournent en simultané :**
  - ∅ Les systèmes d'exploitation *mono tâche*.
  - ∅ Les systèmes d'exploitation *multi tâches* peuvent faire fonctionner plusieurs applications en même temps ;
  - ∅ Les systèmes d'exploitation *préemptifs* gèrent le temps processeur alloué à chaque application. Un commutateur de tâches intervient pour répartir l'allocation des ressources. Des degrés de priorité sont accordés à chaque application. Chaque application peut être interrompue sans interférer avec les autres applications.
  - ∅ Les systèmes d'exploitation *coopératifs*. Une seule application peut monopoliser toutes les ressources de l'ordinateur, et ne rendre la main aux autres applications uniquement quand elle aura terminée...

**Le nombre d'utilisateurs :**

∅ Les systèmes d'exploitation *mono utilisateurs (monoprogrammation)*.

∅ Les systèmes d'exploitation *multi utilisateurs (multiprogrammation)* peuvent supporter plusieurs sessions en même temps.

**La connectivité réseau :**

∅ Les systèmes d'exploitation *clients*,

∅ Les systèmes d'exploitation *serveurs*.

**Le nombre de bits des instructions des programmes qui sont développés pour fonctionner avec tel ou tel système:**

∅ Les applications *16 bits* ;

∅ Les applications *32 bits* ;

∅ Les applications *64 bits ...*

**Le nombre de processeur :**

∅ Les systèmes d'exploitation *mono processeur* ;

∅ Les systèmes d'exploitation *multi processeur* (Windows NT et UNIX) ;

## II.4. NOTION SUR LE NOYAU D'UN SYSTEME D'EXPLOITATION

### II.4.1. DEFINITION

Le **noyau** d'un système d'exploitation est le logiciel qui assure :

- La communication entre les logiciels et le matériel ;
- La gestion des divers logiciels exécutés sur une machine (tâches) : lancement des programmes, ordonnancement, ... ;
- La gestion du matériel : mémoire, processeur, périphérique, stockage, ...

La majorité des systèmes d'exploitation est construite autour de la notion de noyau, c'est-à-dire d'un programme unique responsable de la communication entre le matériel et le logiciel. Outre, le **noyau** des systèmes d'exploitation (appelé **kernel en anglais**), est la partie fondamentale du système d'exploitation. Il gère les ressources de l'ordinateur et permet aux différents composants matériels et logiciels de communiquer entre eux. Windows regroupe dans son noyau un grand nombre de composants permettant au système de pouvoir communiquer avec plusieurs périphériques.

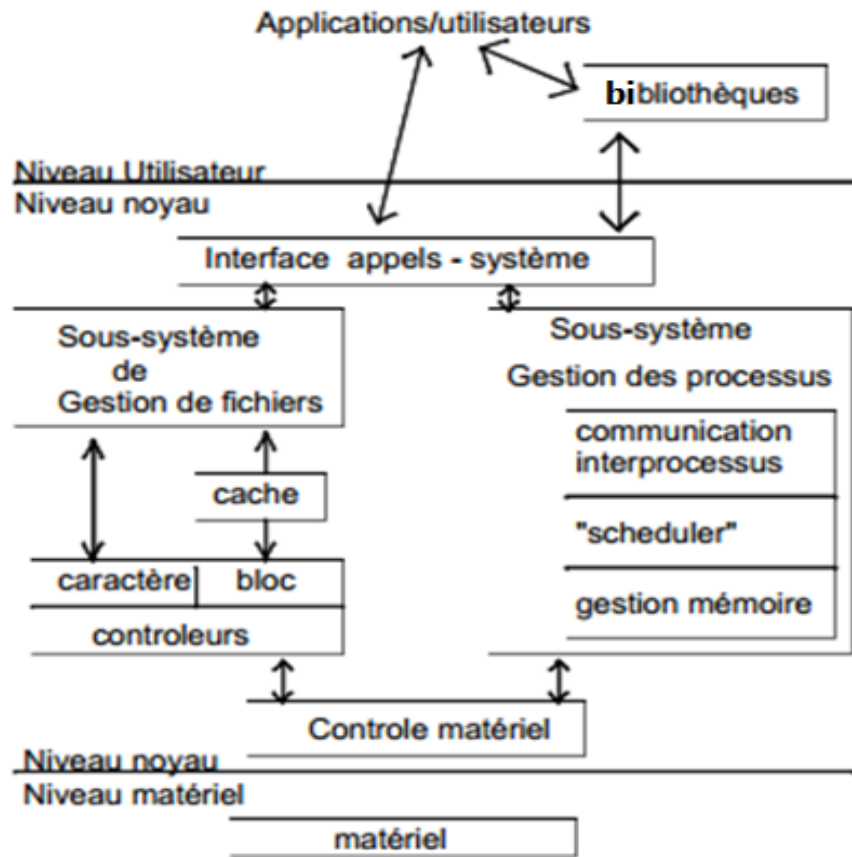


Figure N° 2.1.2: Architecture interne du Noyau (Unix)

L'architecture interne du *Noyau*, c'est à dire l'architecture du programme qui va nous interfacier avec le matériel. Le but ici est de simplifier la compréhension et la fabrication du système. Nous cherchons donc ici à décomposer le noyau en parties disjointes (qui sont concevables et programmables de façons disjointes). Le noyau est généralement exécuté dans un espace mémoire séparé de l'espace des applications: *espace noyau*. Par opposition à *l'espace utilisateur*. Le passage entre ces deux espaces se fait via des appels systèmes. L'intérêt de cette séparation est que le système ne crash/se plante pas si une application plante.

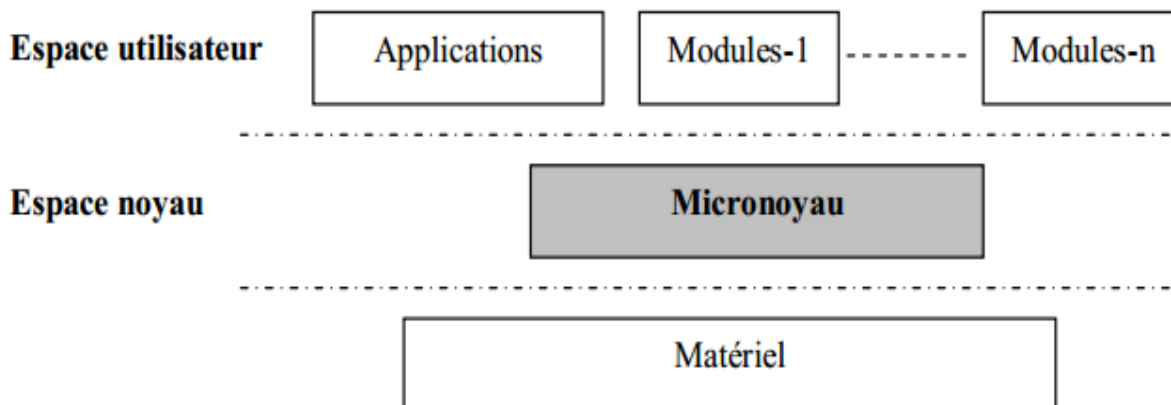


Figure N° 2.1.3: Architecture exécutive du Noyau dans un espace mémoire

## II.4.2. ARCHITECTURES DE DIFFERENTS TYPES DE NOYAUX

### 1) Noyau monolithique

Le *noyau monolithique* utilise un seul bloc contenant l'ensemble des services système (appelé encore *mode noyau*). Il permet de : **Facilité de conception, le Code dur est à maintenir**. Le support par les architectures monolithiques des chargements à chaud ou dynamiques implique une augmentation du nombre de pilotes matériels compilés dans le noyau, et par suite, une augmentation de la taille de l'empreinte mémoire des noyaux. Celle-ci devient rapidement inacceptable. Les multiples dépendances créées entre les différentes fonctions du noyau empêchaient la relecture et la compréhension du code. L'évolution du code s'est faite en parallèle à l'évolution du matériel, et des problèmes de portage ont alors été mis en évidence sur les noyaux monolithiques.

*e.g. Dos, très vieux UNIX et Linux, etc.*

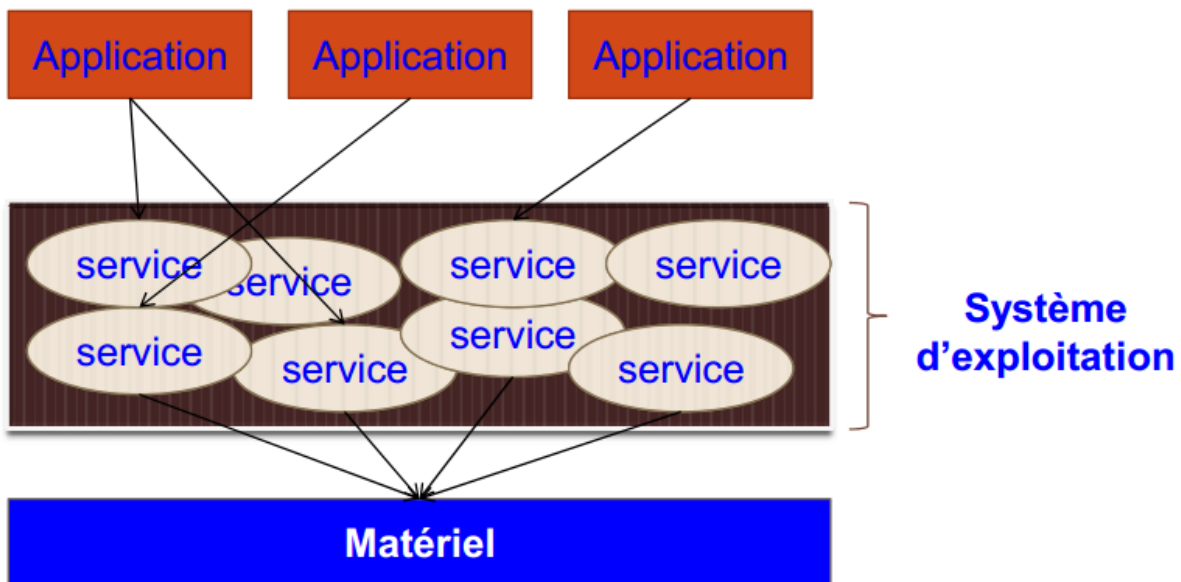


Figure N° 2.1.4: Architecture Noyau monolithique

### 2) Noyau multicouches

Le **noyau multicouches** est organisé en hiérarchie de couches. Chacune construite sur la base des services offerts par la couche inférieure. Il permet de **Facilité de conception et de développement, son code est plus organisé, le chargement des fonctionnalités à la demande**.

Avec ce genre d'organisation, on peut éviter d'avoir à charger l'intégralité des fonctionnalités du noyau et tous les pilotes de périphériques au démarrage de l'ordinateur : on charge ce dont on a besoin quand on en a besoin. Cela permet d'éviter de charger un pilote d'un périphérique qui n'est pas branché sur l'ordinateur par exemple..

**e.g. Linux, BSD, SOLARIS**



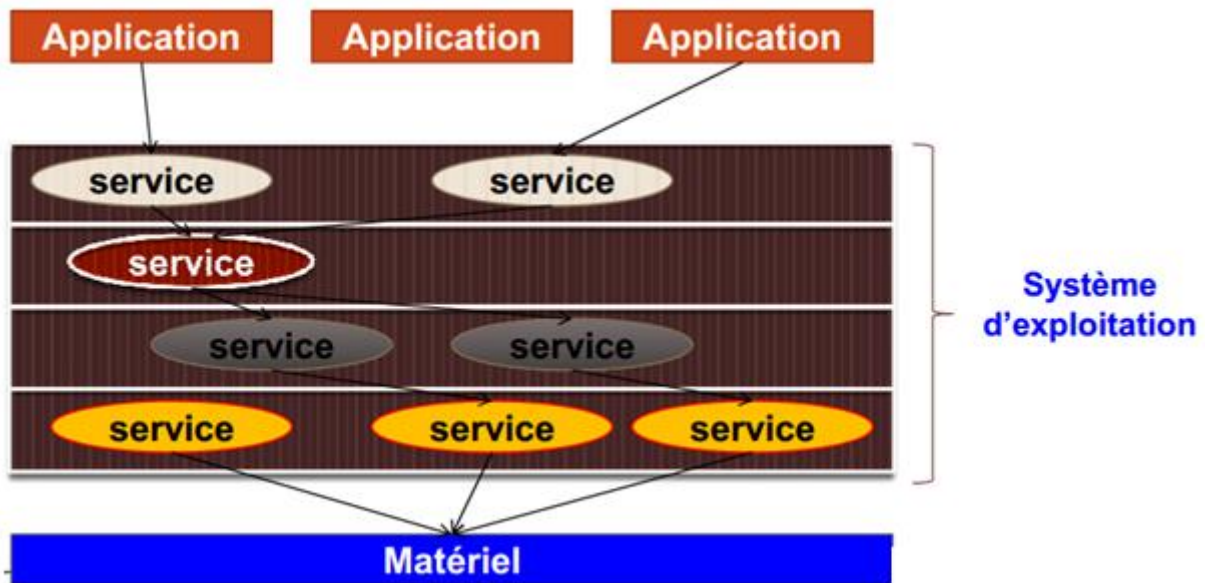


Figure N° 2.1.5: Architecture Noyau multicouches

### 3) Noyau micronoyau

Le **noyau micronoyau** déplace plusieurs fonctions de l'OS vers des « processus serveur » s'exécutant en mode utilisateur réduire la taille du code (en mode noyau). Il a pour rôle : **d'assurer la gestion des communications entre applications et serveurs pour:**

- Φ Renforcer la politique de sécurité,
- Φ Permettre l'exécution de fonctions OS (accès aux registres d'E/S, etc.),

Il a comme caractéristiques :

- ❑ Sa fiabilité est augmentée: si un processus serveur « crash », le système continue à fonctionner et il est possible de relancer ce service sans redémarrer,
- ❑ Son modèle est facilement étendu à des systèmes distribués,

e.g. MAC OS X, GNU HURD, Windows NT

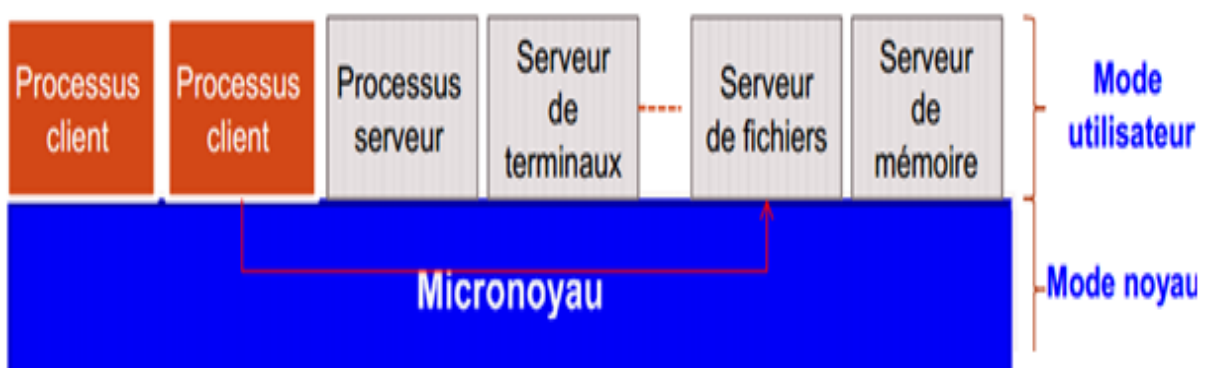


Figure N° 2.1.6: Architecture d'un système à micronoyau

#### 4) Noyau exo-noyau

Le **noyau exo-noyau** se contente de multiplexer et protéger l'accès aux ressources. Tout le traitement qu'on trouve habituellement dans un noyau (mémoire virtuelle, système de fichiers, ...) est délégué à l'espace utilisateur via des bibliothèques appelées «LibOS». Plusieurs LibOS peuvent être utilisées en parallèle.

Il a comme avantages : **gain de performance et une isolation des bugs ;**

e.g. XOK et ExOS du MIT

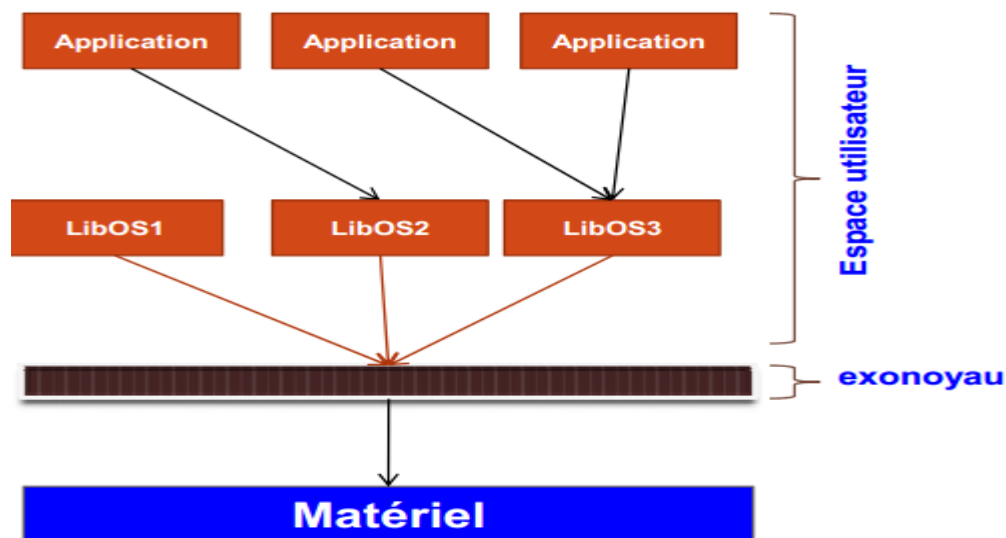


Figure N° 2.1.7: Architecture Noyau exo-noyau

#### 5) Noyau hybride

La construction hybride ressemble à une construction microkernel, cependant certaines fonctions ont été placées dans le noyau pour des raisons d'efficacité. Certains composants tournent en mode privilégié, Un essai de combiner les avantages des deux approches (Windows, MacOS). La dénomination « noyaux hybrides » désigne principalement des noyaux qui reprennent des concepts à la fois des noyaux monolithiques et des micro-noyaux, pour combiner les avantages des deux.

Lorsque, au début des années 1990, les développeurs et concepteurs se sont aperçus des faiblesses des premiers micro-noyaux, certains réintégrèrent diverses fonctionnalités non fondamentales dans le noyau, pour gagner en performance. Les micro-noyaux « purs » semblaient condamnés à l'échec.

Alors que la philosophie générale des systèmes à micro-noyaux est maintenue (seules les fonctions fondamentales sont dans l'espace noyau), certaines fonctions non critiques, mais très génératrices d'appels système, sont réintégrées dans l'espace noyau.

e.g. **Windows NT 2000 et XP (et également les Windows plus récents) ainsi que DragonFly BSD sont en construction hybride.**

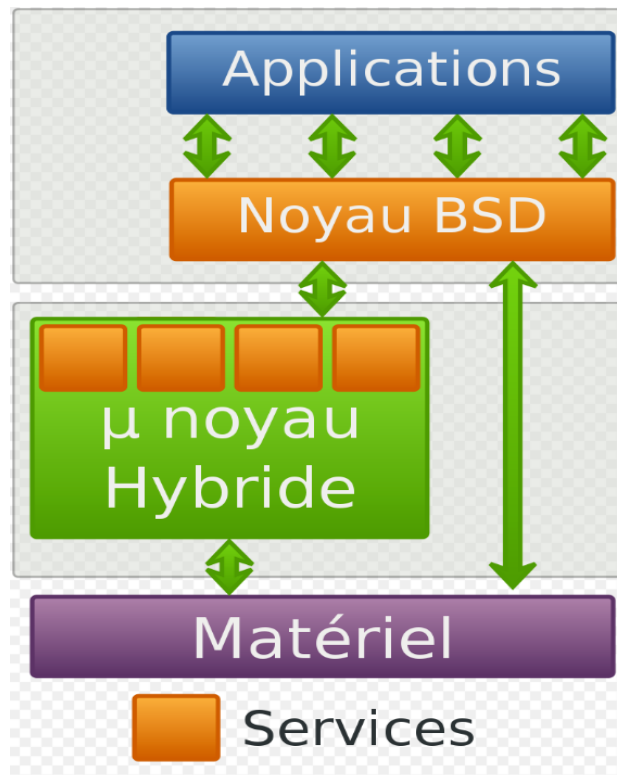


Figure N° 2.1.8: Architecture Noyau hybrides

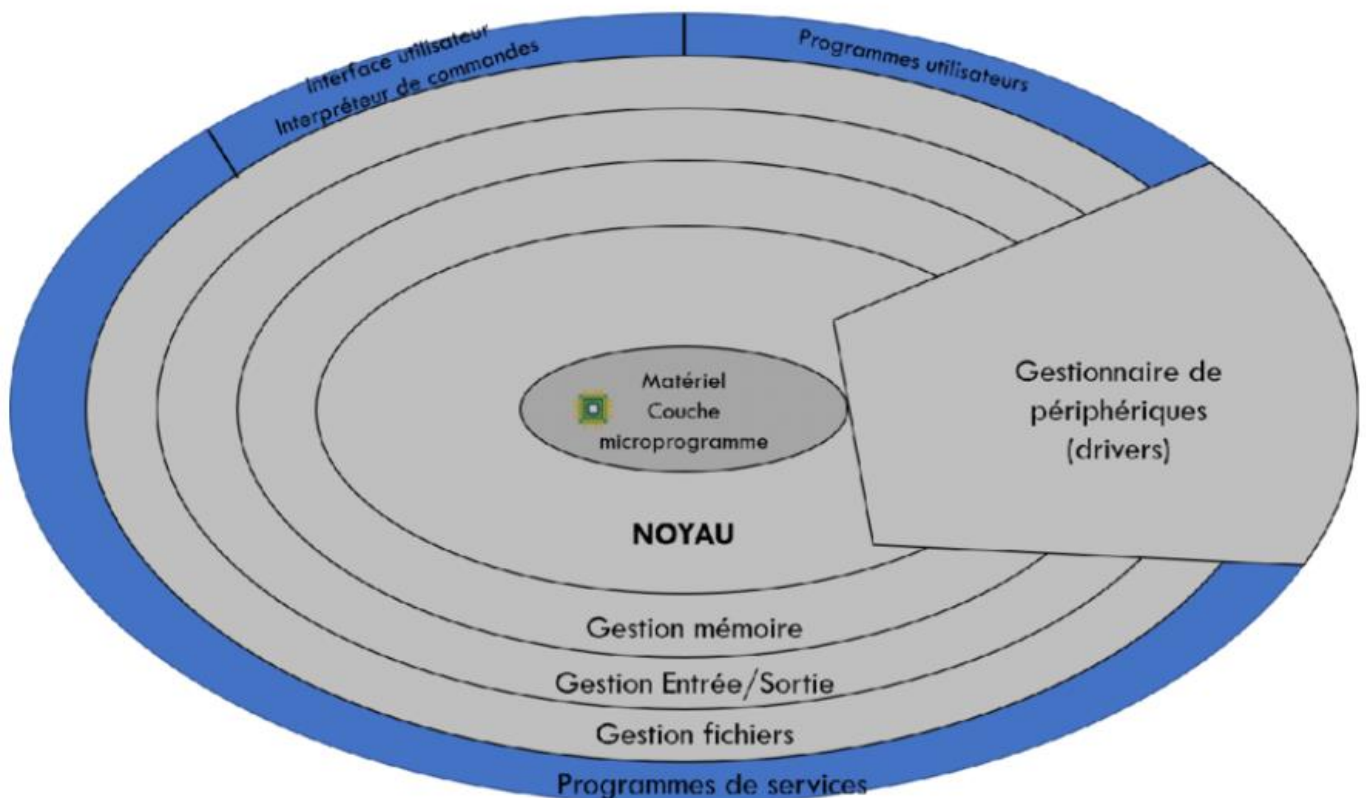


Figure n° 2.1.9 : Structure d'un système d'exploitation

## II.5. PERTINENCE DU SYSTEME D'EXPLOITATION SUR LA VIRTUALISATION

**La virtualisation** est une technique permettant de faire tourner simultanément plusieurs **Systèmes d'Exploitation** sur un même ordinateur physique. C'est ainsi que ces Systèmes d'Exploitation virtuels se partagent les ressources (Processeur, Mémoire, Disques et Interfaces réseau) de la machine physique. Bien plus, pour être utile de manière opérationnelle, la virtualisation doit respecter deux principes fondamentaux :

- ☑ **Le cloisonnement** : Chaque Système d'Exploitation a un fonctionnement indépendant, et ne peut interférer avec les autres en aucun cas ;
- ☑ **La transparence** : Le fait de fonctionner en mode virtualisé ne change rien au fonctionnement du Système d'Exploitation et en occurrence des applications. Ceci implique que toutes les applications peuvent tourner sur un système virtualisé, et leur fonctionnement n'est en rien modifié.

Retenons que plusieurs autres définitions peuvent être considérées, mais l'essentiel a été donné par celle-ci haute. En outre, ce concept fait intervenir la notion de **Machine hôte (Machine physique sur lequel est supporté les OS virtuels)** et **Machine invitée (Machine virtuelle s'exécutant dans l'environnement de virtualisation** : Il est à noter qu'on peut compter plusieurs Machines invitées mais qu'une seule Machine hôte).

La virtualisation système consiste à virtualiser complètement l'environnement matériel c'est-à-dire : Le processeur, la mémoire vive, le disque dur, le réseau et les divers autres périphériques d'entrées/sorties au sein d'une machine virtuelle pour qu'elle puisse accueillir un Système d'Exploitation au complet.

Une machine virtuelle se comporte donc exactement comme un ordinateur physique et contient ses propres ressources matérielles qui sont alors virtuelles (c'est-à-dire basées sur du logiciel). L'autonomie de chaque machine virtuelle rend la solution complètement transparente pour l'utilisateur et toutes actions telles que redémarrage ou installation d'applications ne perturbent pas le fonctionnement des autres machines virtuelles démarrés pour autant sur la même ressource physique. En quelques mots, **la virtualisation permet de rentabiliser au maximum l'utilisation d'une machine physique.**

Une **machine virtuelle (virtual machine)** est une illusion d'un appareil informatique créée par un **logiciel d'émulation**. Le logiciel d'émulation simule la présence de ressources matérielles et logicielles telles que **la mémoire, le processeur, le disque dur, voire le système d'exploitation** et les pilotes, permettant d'exécuter des programmes dans les mêmes conditions que celles de la machine simulée.

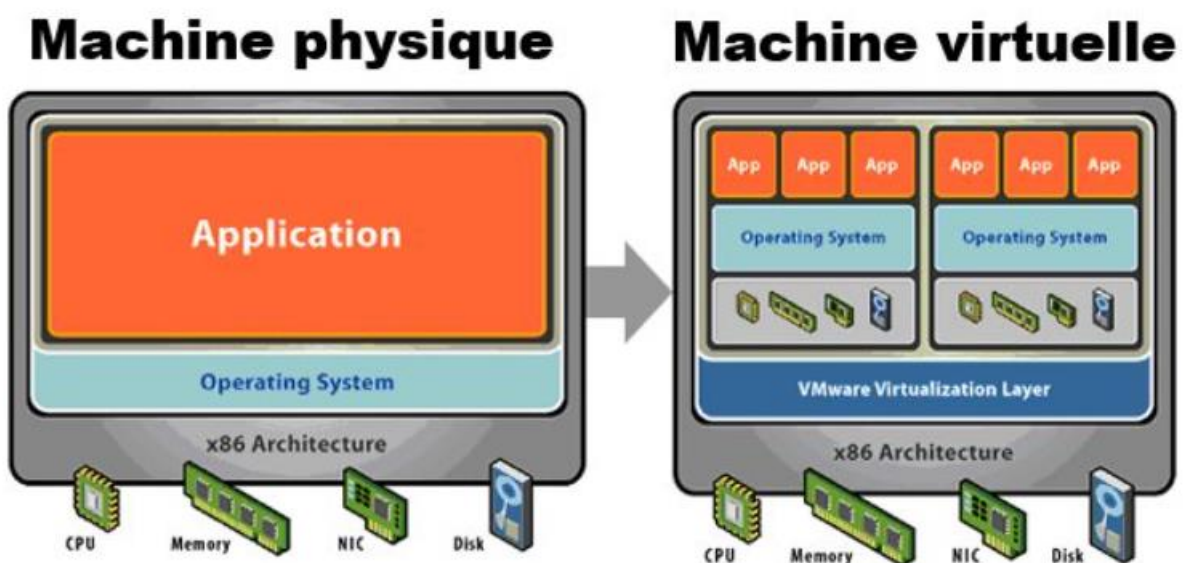
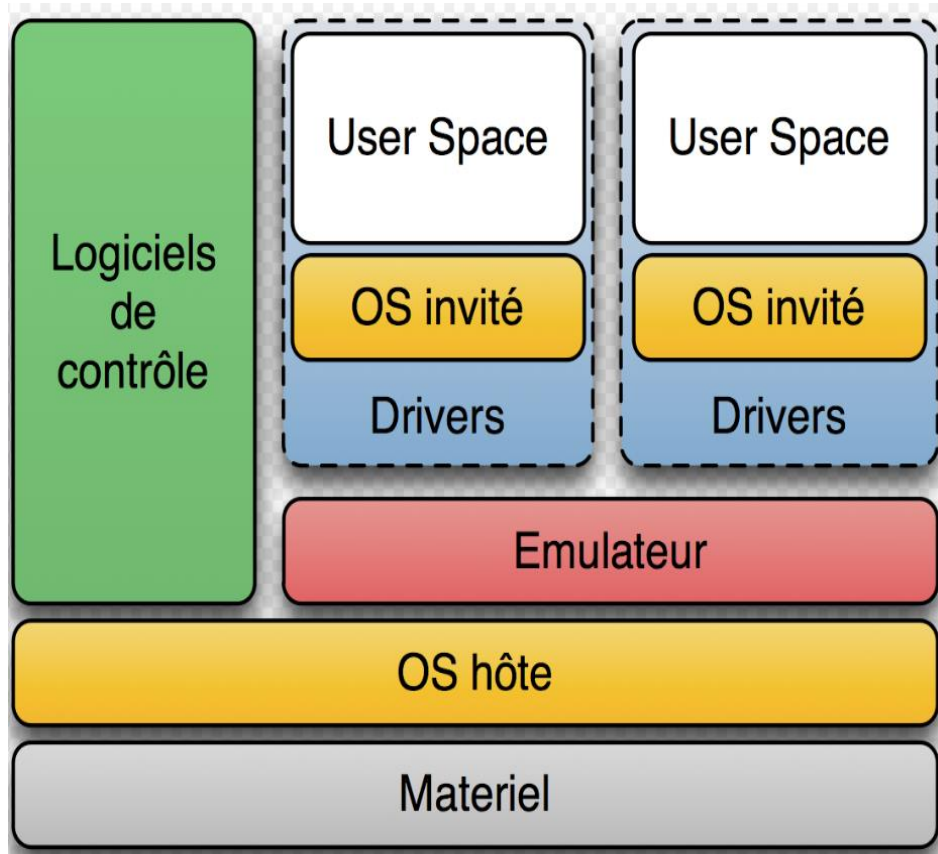


Figure n° 2.1.10 : Architecture de la virtualisation

## II.6. L'ARCHITECTURE CLIENT/SERVEUR

Cette tendance s'est accentuée dans les SE contemporains en tentant de réduire le SE à un noyau minimal. Une des formes les plus accentuées de cette évolution est **l'architecture client/serveur**.

La plupart des fonctionnalités d'un SE sont reportées dans des processus utilisateurs. Pour demander un service comme la lecture d'un bloc de fichier, le processus client envoie une requête à un processus serveur qui effectue le travail et envoie une réponse.

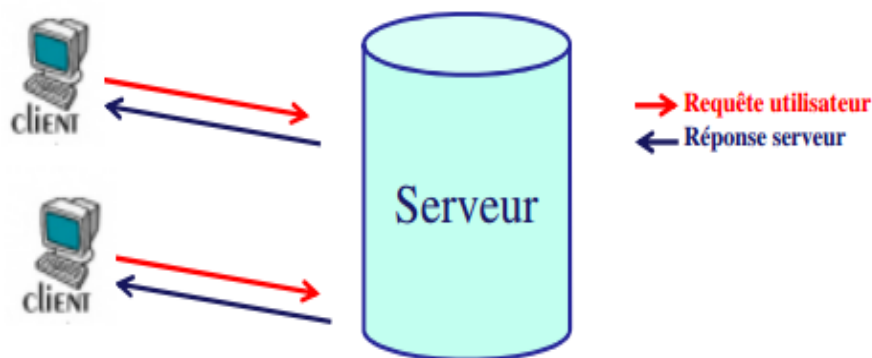


Figure n° 2.1.11 : Architecture client/serveur

Le noyau ne gère que **la communication entre les clients et les serveurs**.

**Remarque :** Le **noyau** est souvent obligé de gérer certains processus serveurs critiques comme les pilotes de périphériques qui adressent directement le matériel.

La décomposition du SE **en modules très spécialisés** le rend facile à modifier. Les **serveurs** s'exécutent comme des **processus en mode utilisateur** et non pas **en mode noyau**. Donc ils n'accèdent pas directement au matériel, ainsi une erreur n'affecte que le serveur et pas l'ensemble de la machine. Ce modèle est bien adapté aux systèmes distribués. Un client n'a pas besoin de savoir si le SE fait exécuter sa requête par un serveur de sa propre machine ou celui d'une machine distante.

## II.7. DEMARRAGE DU SYSTEME D'EXPLOITATION

Lors de ce démarrage, plusieurs étapes sont nécessaires pour permettre de donner la main au noyau du système d'exploitation. Lors du démarrage de la machine, la mémoire principale se trouve dans un état indéterminé. Un programme de démarrage (*bootstrap* en anglais) doit être exécuté pour charger le noyau depuis le disque et démarrer celui-ci. Ce programme de démarrage est généralement stocké dans une mémoire non volatile (souvent dénotée ROM, pour *Read-Only Memory*). Cette mémoire ROM utilise une technologie différente de la mémoire principale, et son contenu n'est pas perdu lors de la mise hors

tension de la machine. En pratique, le type de mémoire utilisé n'est pas seulement en lecture seule (Read-Only) mais supporte des mises à jour occasionnelles nécessitant un programme spécial (on parle alors d'un *firmware*, et d'une mise à jour de *firmware*)

Le processeur reçoit lors du démarrage (ou du redémarrage) de la machine une interruption dite de remise à zéro. Il charge alors son compteur de programme à la première adresse de la mémoire ROM. Cette adresse contient la première instruction du programme de démarrage. Ce dernier va en général effectuer tout d'abord un certain nombre de vérifications de la machine (comme par exemple l'absence d'erreur au niveau de la mémoire principale), initialiser les registres matériels, les bus de communication, et les gestionnaires de périphériques.

Ensuite, ce programme va devoir récupérer sur le disque le code du noyau à proprement parler, pour le copier en mémoire principale et enfin brancher vers sa première instruction. Sur la plupart des systèmes, cette étape se déroule en deux temps : le programme de démarrage est seulement capable de lire le tout premier bloc d'un support de stockage (en général un disque dur ou SSD) dans lequel un programme de chargement plus complet est stocké. C'est ce dernier qui va charger le code du noyau depuis son emplacement effectif sur le disque (le noyau n'est pas stocké dans le premier bloc, mais dans le système de fichier; sous Linux ce fichier est généralement stocké dans le répertoire /boot, par exemple /boot/vmlinuz-3.13.0-32-generic). Sous Linux, le gestionnaire de démarrage GRUB joue ce rôle. Il permet par ailleurs de gérer le démarrage de plusieurs systèmes (comme Solaris, Windows, etc.) ou bien de permettre le démarrage de différents noyaux pour un même système, ce qui est parfois utile pour les développeurs. On notera que lors de l'exécution de GRUB, avant l'exécution du noyau Linux lui-même, les modules de Linux permettant d'utiliser le système de fichier ne sont pas chargés. GRUB inclue donc ses propres modules pour pouvoir utiliser les systèmes de fichiers les plus courants et y localiser le fichier contenant le code du noyau.

## CHAPITRE III : GESTION DES RESSOURCES INFORMATIQUES : LES MEMOIRES, LES PERIPHERIQUES ET LES FICHIERS

La **gestion des ressources informatiques** correspond à la coordination de l'ensemble des ressources, systèmes, plateformes, environnements et équipes informatiques. Pour gérer les environnements informatiques modernes, toujours plus dynamiques, il faut adopter une nouvelle approche qui permet d'améliorer leur rapidité, leur stabilité et leur évolutivité.

### III.1. DEFINITION

En **informatique**, les **ressources sont** des composants, matériels ou logiciels, connectés à un ordinateur. Tout composant de système interne est une **ressource**. Les **ressources** d'un système virtuel incluent les fichiers, les connexions au réseau, et les zones de mémoire.

### III.2. GESTION DE LA MEMOIRE

La **gestion de la mémoire** est une forme de gestion des ressources appliquée à la mémoire de l'ordinateur. L'exigence essentielle de la gestion de la mémoire est de fournir des moyens d'allouer dynamiquement des portions de mémoire aux programmes à leur demande, et de les libérer pour réutilisation lorsqu'elles ne sont plus nécessaires. Ceci est essentiel pour tout système informatique avancé où plus d'un **processus** peuvent être en cours à tout moment.

Le gestionnaire de mémoire est le sous-ensemble du système d'exploitation qui permet de gérer la mémoire de l'ordinateur. Sa tâche la plus basique est d'allouer de la mémoire à des processus lorsqu'ils en ont besoin. Cette mémoire allouée est par défaut propre au processus qui en fait la demande. La mémoire physique sur un système se divise en deux catégories :

- ∅ **Les mémoires conventionnelles**: composées de circuit intégrés, donc très rapide (**RAM et ROM**) ;
- ∅ **Les mémoires de masses** : composée de supports magnétiques (**disque dur, bandes magnétiques...**), beaucoup plus lente ;

Le **gestionnaire de mémoire** est un sous-ensemble du système d'exploitation. **Son rôle est de partager la mémoire entre l'OS et les diverses applications**. Le terme "**mémoire**" fait surtout référence la mémoire principale, **c'est à dire à la RAM**, mais la gestion de celle-ci demande la contribution de la mémoire auxiliaire (mémoire de masse, spacieuse mais lente) et à la mémoire cache (rapide mais de taille restreinte).  
La gestionnaire de mémoire assure 4 fonctions à savoir :



- ☑ **L'allocation de la mémoire aux processus :** (Répertorier les emplacements libres de la mémoire ; Allouer la mémoire nécessaire aux nouveaux processus ; Récupérer la mémoire des processus qui s'achèvent). Cette récupération peut nécessiter une réallocation des processus en cours pour optimiser l'emploi de la mémoire. La zone mémoire attribuée à un processus peut donc changer au cours de son exécution.
- ☑ **La protection :** Il faut s'assurer que les adresses générées par chaque processus ne concerne que la zone mémoire qui lui est impartie, sans quoi, l'intégrité du système d'exploitation et des autres processus n'est pas garantie. Certaines zones mémoire doivent pourtant servir simultanément à plusieurs processus : le code de fonctions servant à plusieurs applications qui tournent en parallèle ou les données utilisées simultanément par divers processus.
- ☑ **La segmentation de l'espace d'adressage :** Les programmes sont subdivisés en segments : le code, les données modifiables, celles qui ne le sont pas, la pile. On attend donc du gestionnaire de mémoire qu'il permette la segmentation de l'espace d'adressage des programmes pour les raisons suivantes : (pouvoir coder les segments séparément et les paramétrer en fonction de l'application, permettre des degrés de protection différents selon les segments(lecture seule, exécution...), accepter le partage de certains segments).
- ☑ **La mémoire virtuelle :** Elle offre aux applications une mémoire de taille supérieure à celle de la mémoire principale. L'espace d'adressage qu'offre la mémoire centrale est parfois insuffisant. Les disques suppléent à cette insuffisance en fournissant une mémoire auxiliaire plus vaste mais plus lente et qui n'est pas directement accessible au processeur.

### III.2.1. Gestion de la mémoire pour systèmes monotâches

Dans le cas des systèmes monotâche, la gestion de la mémoire est assez simple. Il suffit de réserver une partie de la mémoire au système d'exploitation. L'application est ensuite casée dans l'espace restant qui est libéré sitôt que l'application est terminée. Cela se complique un peu si l'application nécessite plus d'espace que ce que peut fournir la mémoire vive.

On segmente alors l'application en segments de recouvrements ou "Overlays". Cette technique n'a plus cours maintenant. Elle était utilisée à l'époque du DOS pour des applications volumineuses. Le programmeur devait prévoir le découpage de son application en imaginant comment ces overlays serait chargés en mémoire les uns à la suite des autres pour qu'au cours de son exécution l'application puisse atteindre toutes les fonctions nécessaires.

C'était en quelque sorte comme cela qu'on concevait la mémoire virtuelle à l'époque des systèmes d'exploitation monotâche. Notez que la zone mémoire occupée par le système d'exploitation n'était pas protégée par ce type de gestion de mémoire.

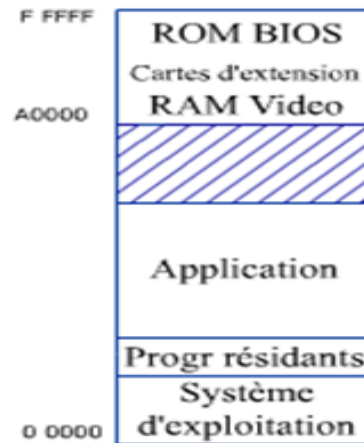


Figure n° 3.1.0: Architecture mémoire système monotâches

### III.2.2. Gestion de la mémoire pour systèmes multitâches

Plusieurs processus doivent se partager la mémoire sans empiéter sur l'espace réservé au système d'exploitation ni aux autres processus. Quand un processus se termine, le S.E. doit libérer l'espace mémoire qui lui était alloué pour pouvoir y placer de nouveaux processus.

#### Partition de la mémoire

##### a) Partitions fixes

Le plus simple est de diviser la mémoire en partitions fixes dès le démarrage du système. Les partitions sont de différentes tailles pour éviter que de grandes partitions ne soient occupées que par de petits processus. Le gestionnaire de mémoire, en fonction de la taille des processus, décide quelle partition lui allouer pour ne pas gaspiller trop de mémoire. Une file d'attente est associée à chaque partition. Quand vient une nouvelle tâche, le gestionnaire détermine quelle est la plus petite partition qui peut la contenir puis place cette tâche dans la file correspondante.

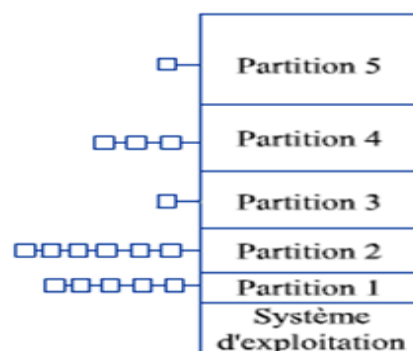


Figure n° 3.1.1: Architecture partition de la mémoire

Le fait d'éviter d'allouer une partition trop grande à un petit processus conduit parfois à des aberrations. Il arrive que des partitions plus grandes restent inutilisées alors que se forment ailleurs des files interminables de petits processus. La mémoire est donc mal utilisée. Une autre solution est de créer une file unique. Lorsqu'une partition se libère, on consulte la file pour trouver la tâche qui l'occuperait de manière optimale. Le risque est que les petites tâches soient pénalisées. Une parade est de conserver une petite partition au moins qui ne sera accessible qu'aux petites tâches. Une autre solution, serait de dire qu'un processus ne peut être ignoré qu'au maximum un certain nombre de fois. Après  $n$  refus, il prendra place dans une partition même si la partition est bien plus grande qu'il ne faut.

#### b) Partitions variables

Une autre manière d'éviter les emplacements mémoires inoccupés en fin de partitions est d'allouer aux processus des espaces qui correspondent exactement à l'espace qui leur est utile. Au fur et à mesure que les processus se créent et se terminent, des partitions s'allouent et se libèrent laissant des zones mémoires morcelées et inutilisables. La mémoire se fragmente et est de plus en plus mal employée. Il faudrait la compacter en déplaçant régulièrement les processus mais cette tâche supplémentaire ralentit le système. Le partitionnement de la mémoire que ce soit avec des partitions de tailles fixes ou de tailles variables, ne permet pas d'utiliser la mémoire au mieux.

### III.2.3. La pagination

Les processus requièrent des espaces d'adresses continus. On a vu que cela est difficilement réalisable en découpant la mémoire en partitions dont les tailles correspondent à celles des processus. La pagination est une technique d'allocation de la mémoire bien plus efficace. Elle fournit aux processus des **espaces d'adresses** séquentiels à partir d'**espaces mémoire** discontinus.

**La pagination** consiste à diviser la mémoire et les processus en blocs de mêmes tailles appelés pages. Les **pages mémoire** sont souvent appelées "frames" ou "cadres" tandis que les **pages de processus** sont simplement appelées "pages".

Les pages (de processus) ne sont pas toutes simultanément actives ; elles ne sont donc pas nécessairement toutes présentes simultanément dans la mémoire principale. Les pages inactives attendent sur le disque. L'**espace d'adressage** est donc **virtuel** sa taille peut être supérieure à celle de la mémoire réelle.

Les processeurs disposent actuellement d'un dispositif, le **MMU "Memory Manager Unit"** qui permet de placer des processus en mémoire sans nécessairement placer les pages de processus dans des cadres de pages contigus. On distingue les adresses logiques qui se réfèrent aux pages de processus des adresses physiques qui se réfèrent aux cadres de pages.

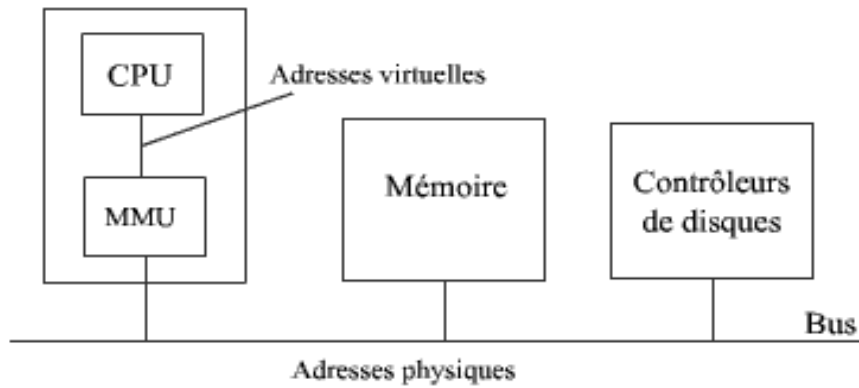


Figure n° 3.1.2: Architecture de l'Unité de Gestion de la Mémoire (MMU/Memory Management Unit)

Voyons à présent comment l'unité de gestion mémoire (MMU) met en correspondance les adresses physiques et logiques. Elle contient pour ce faire une **table de pages** où sont inscrits les numéros des cadres de pages.

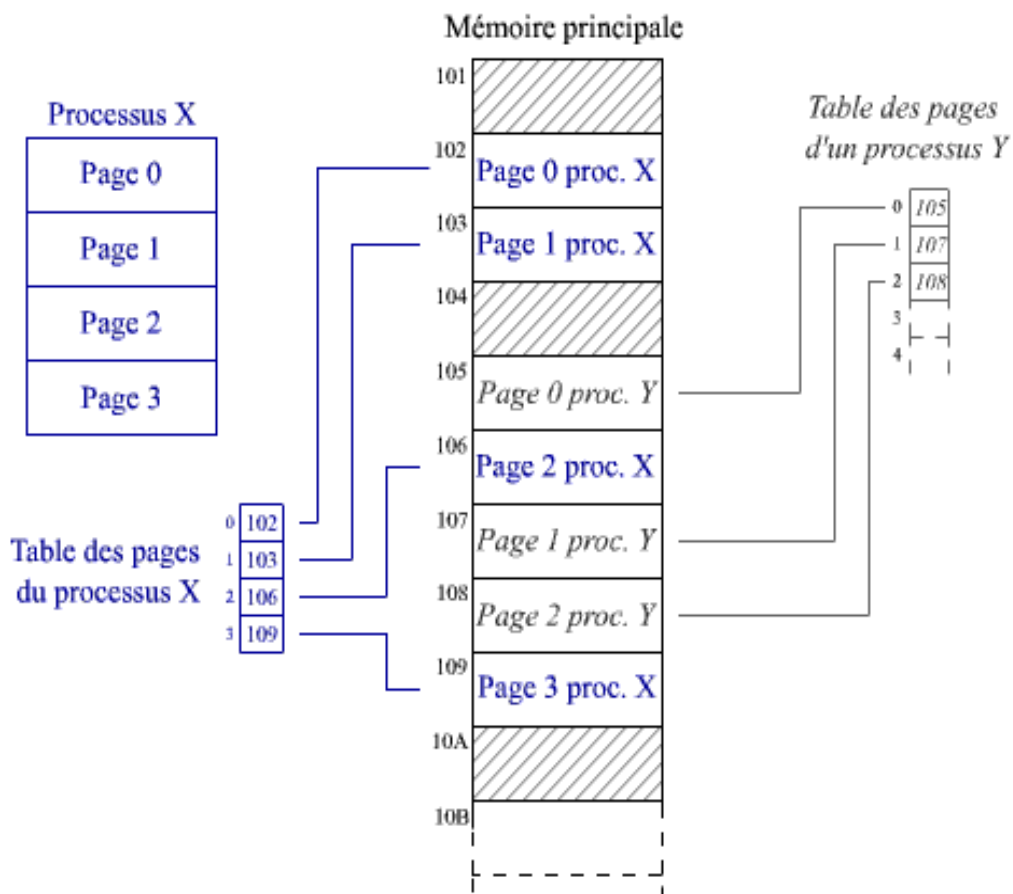


Figure n° 3.1.3: Architecture de la table de pages des adresses physiques et logiques

### a) Fonctionnement des tables de pages

L'adressage se fait au moyen de numéros de pages et d'offsets. L'offset (= déplacement ou décalage) est la position relative au début de la page. L'adresse logique est composée du numéro de page de processus et d'un offset. L'adresse physique correspondante est formée à partir du numéro du cadre de page où est chargé la page de processus et du même offset que celui de l'adresse logique. Le numéro du cadre de page est consigné dans une table des pages associée au processus. On y retrouve le numéro du cadre de page en se servant du numéro de page de processus comme d'un index.

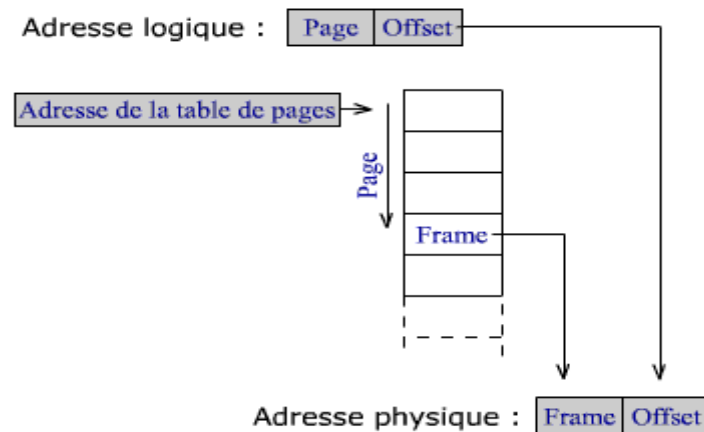


Figure n° 3.1.4: Architecture de fonctionnements des tables de pages

### b) Pagination simple

Le nombre de pages étant souvent très grand les tables des pages deviennent volumineuses et peuvent même occuper ... plusieurs pages. On les fractionne donc en plusieurs niveaux : une table de page de niveau supérieur dont chaque élément pointe vers une table de niveau inférieur. L'adresse logique contient dès lors deux nombres pour aboutir au numéro de page. Le premier sert d'index dans la table de niveau supérieur, le second sert d'index dans la table du niveau suivant.

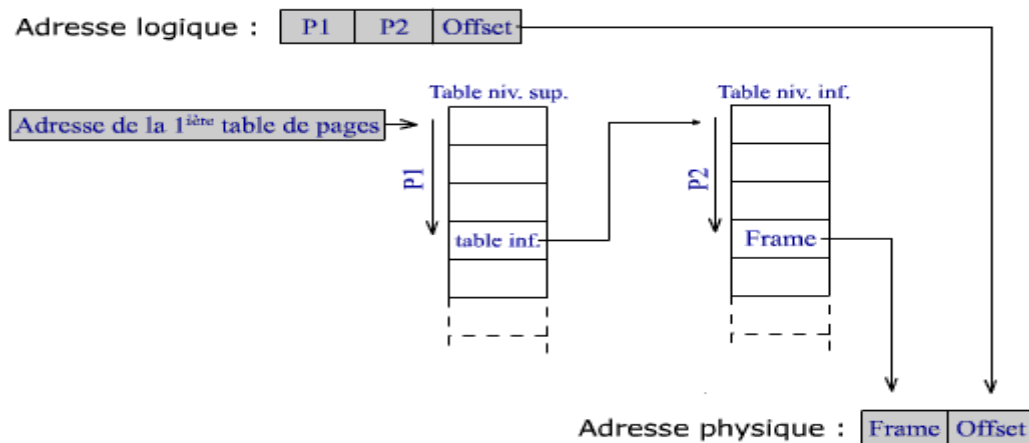


Figure n° 3.1.5: Architecture de la pagination de la table de pages

### c) Tables de pages multiniveaux

Ces accès multiples à différentes pages pour aboutir à l'adresse finale ralentissent fortement l'adressage. On évite de répéter ces recherches en notant les correspondances trouvées entre les adresses logiques et les adresses physiques dans une mémoire associative. Ce qui permet ensuite de retrouver presque immédiatement les correspondances les plus récentes.

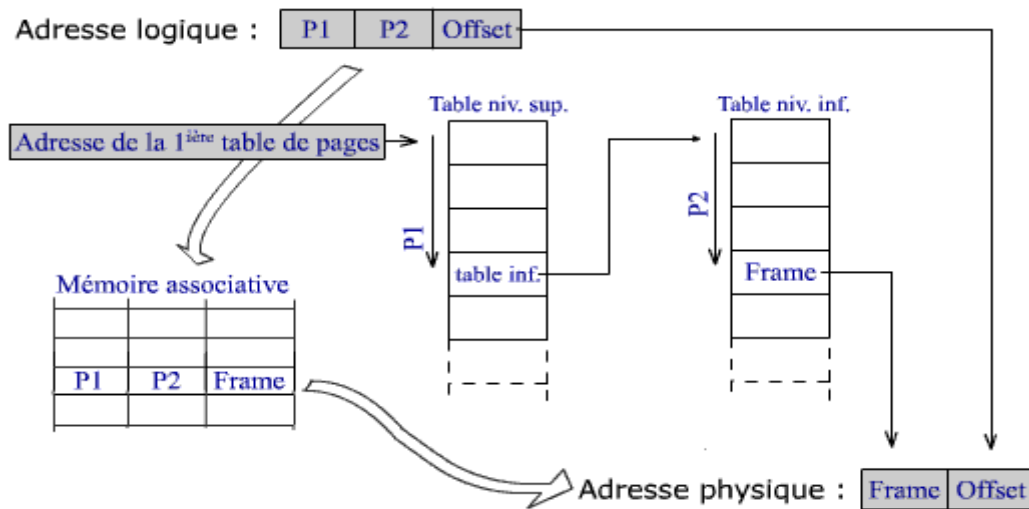


Figure n° 3.1.6: Architecture de la pagination de la table de pages

**Remarque :** L'espace d'adressage est perçu par le programmeur comme une suite continue d'octets. La subdivision de l'adresse en numéros de page et d'offset est transparente. Elle est prise en charge par le matériel.

### III.2.4. La segmentation

Chaque processus est constitué d'un ensemble de segments. Chaque segment est un espace linéaire. Les segments sont des espaces d'adressages indépendants de différentes longueurs et qui peuvent même varier en cours d'utilisation. Ils correspondent à des **subdivisions logiques déterminées par le programmeur ou par le compilateur**. Les segments contiennent des **informations de même nature** : le code, les données, la pile, des tables, etc. Il est dès lors possible d'attribuer des protections adaptées à chaque type de segment : un segment de code peut être déclaré en exécution seule, une table de constantes en lecture seule mais pas en écriture ni en exécution. Certaines zones de code en exécution seule peuvent être partagées par plusieurs processus ; cela se fait par exemple pour des bibliothèques de sous-programmes. L'accès aux segments se fait via une **table de segments**.

Chaque entrée de la table comporte l'**adresse de départ** du segment et sa **taille**. L'adresse logique est constituée du numéro de segment et d'un offset. Contrairement aux pages dont le fonctionnement est

transparent pour le programmeur, les segments sont des entités logiques qu'il connaît et manipule. Il distingue les deux informations contenues dans l'adresse : le numéro du segment et l'offset. Le numéro de segment sert d'index pour retrouver l'adresse du début du segment dans la table de segment. Cet offset doit être inférieur à la taille du segment consignée elle aussi dans la table de segment. Si ce n'est pas le cas, une erreur est générée qui provoque l'abandon du programme. L'offset est ensuite ajouté à l'adresse de début de segment pour former l'adresse physique.

### III.2.5. Segmentation avec pagination

La segmentation et la pagination concernent des problèmes différents. Ce sont deux techniques qui peuvent se combiner :

- ∅ La segmentation découpe les processus en zones linéaires pouvant être gérées différemment selon que ces segments sont propres au processus, qu'ils sont partagés, lus, écrits ou exécutés et de manière à protéger les processus entre eux.
- ∅ La pagination découpe la mémoire en pages non contiguës mais de même taille. Elle procure aux processus des espaces d'adresse continus (nécessaires aux segments). Les pages mémoires peuvent n'être allouées que lorsqu'un processus en a besoin. On obtient de la sorte une mémoire virtuelle de taille supérieure à la mémoire réelle.

Les systèmes d'exploitation qui gèrent ces deux techniques simultanément administrent **une table de segments et plusieurs tables de pages**. Un segment peut contenir plusieurs pages mais toutes ne doivent pas nécessairement être présentes en mémoire à tout moment. On ne garde en mémoire que celles qui sont réellement utilisées. Chaque segment a sa propre table de pages (le cas pratique est celui du système Microsoft Windows qu'un fichier appelé "*win386.swp*" d'une taille conséquente, proportionnelle aux besoins en mémoire vive, fait son apparition).

## III.3. GESTION DES PERIPHERIQUES (devmgmt.msc)

Un périphérique informatique fait par définition référence un dispositif connecté à un système de traitement de l'information central, comme un ordinateur par exemple, et a pour fonction principale d'offrir à ce système central des fonctionnalités supplémentaires. Ainsi, un ordinateur peut très bien calculer et traiter tout seul des informations, mais avec des périphériques, il peut partager ces résultats et accepter le traitement et le partage de données supplémentaires. Sans un moniteur par exemple, qui pour ceux qui l'ignorent est bien un périphérique informatique, on ne peut pas voir ce qui est fait par l'ordinateur. Et sans une souris qui déplace le curseur vers différents points de l'écran, on ne peut faire aucune tâche, opération et commande de manière optimale. Il existe de nombreux types de périphérique informatiques :

- ☑ **Les périphériques d'entrée** tels que le clavier, le micro, la webcam, la souris ou la tablette à dessin. Ils sont conçus pour vous permettre de partager des données.
- ☑ **Les périphériques de sortie** tels que les imprimantes, les moniteurs ou encore les hauts parleurs. Sans eux, rien ne sort d'un ordinateur.
- ☑ **Et enfin, les périphériques d'entrée/sortie** tels que les disques durs externes, les clés USB, les lecteurs de CD/DVD, les routeurs, les commutateurs, les répéteurs, les contrôleurs WLAN, les cartes réseau (NIC) et bien d'autres encore qui fournissent à la fois une entrée à votre machine et reçoivent aussi des données.

### III.3.1. Notion

La gestion des périphériques représente peut-être le défi le plus considérable d'un système d'exploitation. Ce dernier doit contrôler tout un ensemble de périphériques avec des différences multidimensionnelles. *Rapidité du périphérique, volume des informations, service proposé, direction du flux d'informations et protocoles de communications* sont autant de grandeurs aux éventails très larges. Outre cette diversité, le système d'exploitation doit pouvoir traiter un grand nombre de périphériques, ce traitement doit se dérouler dans un environnement parallélisé. Les périphériques agissent en général indépendamment de l'UC, en fonction de leur propre fréquence et synchronisation. Le système d'exploitation, qui la plupart du temps s'exécute sur une seule UC, doit donc gérer des requêtes simultanées en provenance d'un grand nombre de périphérique.

### III.3.2. Organisation des dispositifs d'E/S

Même si certains ordinateurs sont différents dans les détails, ils sont conçus autour de la même philosophie. Les dispositifs d'E/S, le mémoire et l'UC communiquent par le biais d'un bus de communication. Les machines les plus simples présentent un seul bus de communication. Mais les communications ne peuvent avoir lieu qu'entre deux éléments à la fois. Un dispositif appelé « Arbitre de bus » décide quel périphérique est autorisé à communiquer au prochain cycle. Celui-là peut communiquer avec n'importe quel autre de son choix.

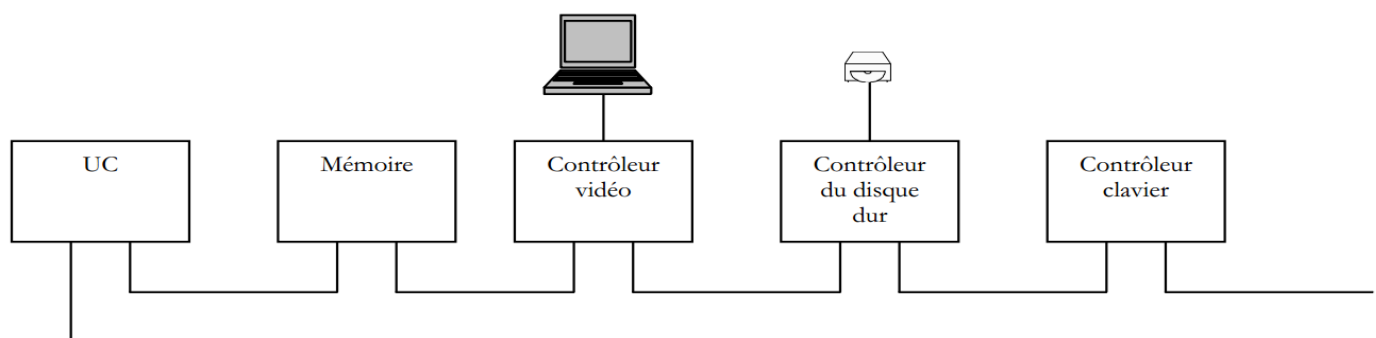


Figure n° 3.1.7: Architecture des dispositifs d'E/S



En principe, le bus est attribué à l'UC afin qu'elle puisse communiquer avec la mémoire. Des accès fréquents à la mémoire et une vitesse relativement rapide de l'UC conduisent à une utilisation élevée du bus par cette dernière. Bien que le bus leur soit fréquemment nécessaire, les E/S ont des besoins en communication généralement plus urgents que les requêtes de l'UC. C'est pourquoi les requêtes des périphériques d'E/S reçoivent souvent une priorité plus élevée. Le processus consistant à retirer le bus de l'UC pour l'attribuer à un périphérique est appelé vol de cycle.

On peut trouver des bus multiples sur des machines pour des raisons de parallélisme et d'ajustement des performances. Les bus multiples permettent à plusieurs communications de se dérouler simultanément. L'UC peut par exemple communiquer avec un port série sur un bus alors qu'un disque communique avec la mémoire sur un autre. Cependant l'avantage des bus multiples est assez limité. La plupart des communications impliquent soit la mémoire, soit l'UC. Sans matériel multiaccès particulier, ils ne peuvent communiquer qu'avec un seul dispositif à la fois. Les architectures PC les plus récentes ont souvent recours à 3 types de bus outre celui du processeur de l'UC : le bus standard de connexion des périphériques, bus PCI. Par ailleurs, un bus mémoire spécial permet des communications optimisées entre l'UC et la mémoire ; un bus ISA (Industry standard architecture) est relié au bus PCI pour offrir une compatibilité descendante pour les anciens périphériques.

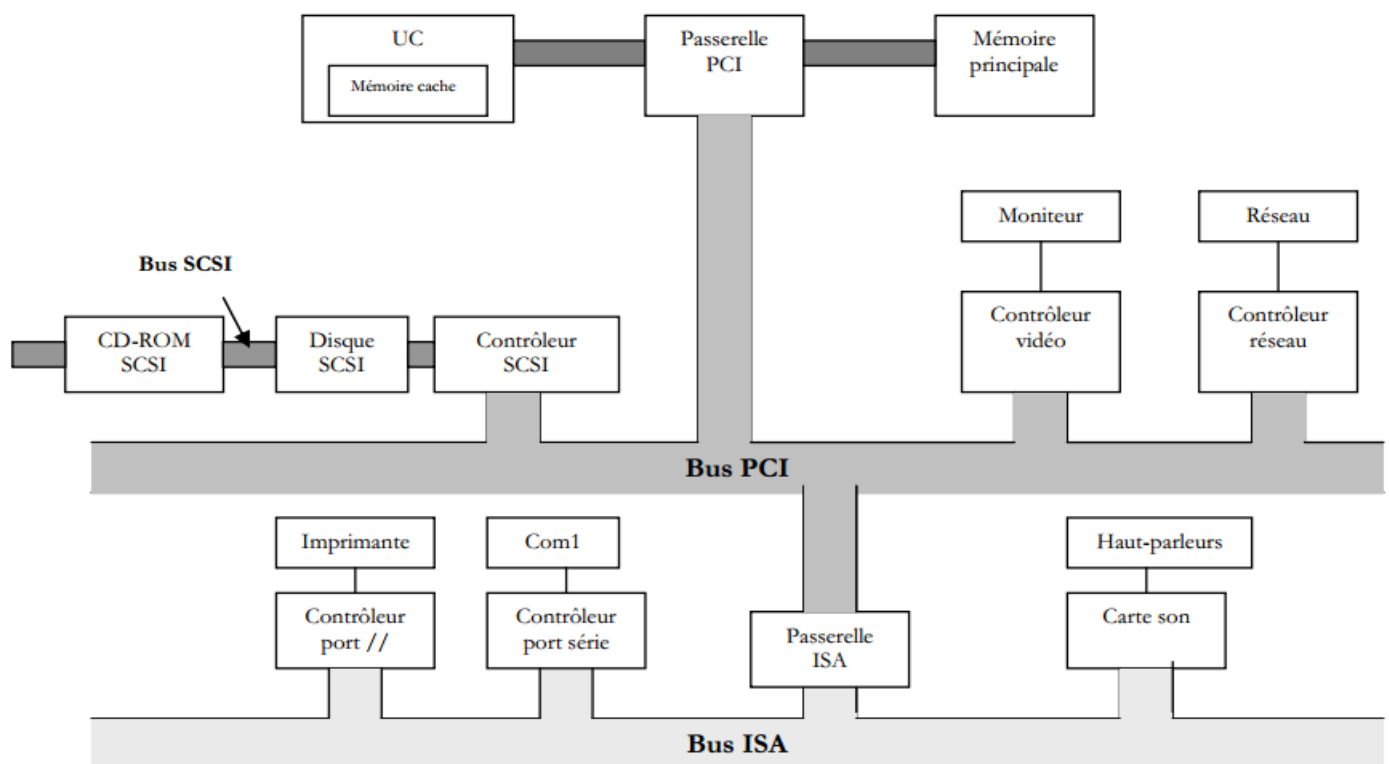


Figure n° 3.1.8: Architecture à bus simple

### III.3.3. Contrôle des E/S

Dans le modèle le plus simple l'UC communique directement avec les périphériques d'E/S et prend en charge le contrôle des moindres détails de l'opération du périphérique. Ce type de communication est de plus en plus rare (encore dans les systèmes embarqués).

Les nouveaux systèmes incorporent la notion de « contrôleur de périphériques ». Une commande classique de l'UC au contrôleur peut être le lancement d'une opération de lecture pour un octet d'informations depuis un appareil en série ou d'un secteur d'informations depuis un disque. Le contrôleur de périphérique transmet au périphérique les commandes plus détaillées nécessaires à la réalisation de l'opération requise. En déchargeant cette responsabilité sur le contrôleur, l'UC est libre d'accomplir simultanément d'autres tâches. Chaque dispositif d'E/S possède un contrôleur spécifique. La plupart des contrôleurs peuvent servir plusieurs périphériques à la fois.

### III.3.4. Ports d'E/S

Pour réaliser les E/S, l'UC doit communiquer avec les modules d'E/S, qu'il s'agisse d'un périphérique ou d'un contrôleur ou d'un canal. Chaque module d'E/S contient un ou plusieurs registres servant à la communication avec le processeur.

En écrivant dans ces registres, le SE ordonne au périphérique de délivrer des données, d'en accepter, de s'activer, désactiver ou effectuer une opération donnée (commande de périphérique). En lisant les registres, le SE connaît l'état du périphérique. De nombreux périphériques sont équipés d'un tampon de données que le SE peut écrire ou lire. Par exemple, la RAM vidéo contient les pixels affichés à l'écran. Cette RAM vidéo est le tampon de données relatif au périphérique vidéo (carte graphique).

### III.3.5. Communication entre UC et E/S

La communication entre les modules d'E/S et l'UC suivent l'un des quatre protocoles suivant :

- ☑ **Attente active** : L'UC émet une commande au module d'E/S pour lancer une opération d'E/S. L'UC entre dans une boucle pour vérifier si l'opération est achevée ou non et ensuite tester si l'opération est finie avec succès ou non. Les processus sont bloqués jusqu'à ce que l'E/S finisse.
- ☑ **La scrutation** : L'UC lance les opérations d'E/S puis retourne pour exécuter les processus du système. Après des délais périodiques, l'UC fait une scrutation des périphériques d'E/S pour voir si l'un d'eux a fini sa tâche ou non. Ce protocole est plus efficace que l'attente active, mais il est plus couteux.
- ☑ **Interruption** : Dans ce cas, le périphérique d'E/S se charge d'informer l'UC de l'achèvement de l'E/S qu'il prépare. L'interruption utilisée par un module est généralement configurable et unique ce qui offre le parallélisme.

- ☑ **Accès direct à la mémoire** : Les composants d'accès direct à la mémoire (DMA : Direct Memory Access) permettent aux modules d'E/S de lire ou écrire des données directement depuis le mémoire. Avec cette technique, les données ne doivent pas transiter par l'UC. Cet accès est utile pour des dispositifs tels que les disques.

### III.3.6. Les pilotes des périphériques

Les pilotes constituent la partie logicielles qui contrôle et interagit directement avec le périphérique d'E/S. Le SE peut exiger que pour chaque périphérique, il existe un ensemble de fonctions mises en œuvre, telle que :

- Φ **Open**: réalise les tâches de démarrage avant l'accès au périphérique ;
- Φ **Close**: fermer le périphérique après l'utilisation ;
- Φ **Schedule**: ordonnance une requête d'E/S ou plusieurs avant d'être passés au périphérique (pilote...) ;
- Φ **Startio**: vérifie si le périphérique est actif ; si ce n'est pas le cas, lance la prochaine opération d'E/S sur la file d'ordonnancement du périphérique ;
- Φ **Interrupt**: routine exécutée lorsque le périphérique envoie une interruption à l'UC ;

Compte tenu des différents types de périphériques pouvant être connectés à certains ordinateurs, il ne serait pas pratique d'inclure les pilotes de tous les périphériques éventuels dans le SE. Les SE doivent être configurés en fonctions des périphériques. Ceci se fait par ajout ou suppression de périphériques (modifier l'image du SE).

### III.3.7. Les périphériques

#### 1) Les périphériques graphiques

L'un des plus grands problèmes est le volume d'informations qui doit être transmis pour décrire l'affichage sur les moniteurs. L'image sur les moniteurs est affichée sous formes de points nommés pixels. Les moniteurs diffèrent selon le nombre de pixels et les couleurs qu'ils peuvent adopter. Les premiers moniteurs étaient composés de 200 lignes de 320 pixels noir et blanc. De nos jours, les résolutions de 800×600, 1024×768 et 1280×960 sont les plus répandues. L'illumination de chaque pixel est contrôlée par des valeurs stockées en mémoire vidéo :

- Φ Les affichages monochromes de base n'ont besoin que d'un bit par pixel.
- Φ Les affichages par niveaux de gris (255 niveaux) ont besoin de 8 bits.

Les affichages couleurs réelles ont besoin de 24 bits : chaque pixels est généré par la combinaison des trois couleurs primaires (rouge, vert et bleu : RGB ou RVB) ; 8 bits pour chacune de ces couleurs. Avec un affichage couleurs réelles et pour un moniteur vidéo 1024×768, il nous faut 2,3 Mo pour stocker une valeur de 24 bits pour chaque pixel.

Pour modifier une image à l'écran, de nouvelles données doivent être écrites en mémoire vidéo. Compte tenu du grand volume de données multimédia cela peut représenter une charge importante sur le système. Par exemple, une animation vidéo qui nécessite 25 image/seconde et 2,3 Mo par image réécrit 25 fois dans la mémoire vidéo par seconde donc un total de 58 Mo/seconde. Cette exigence de transfert a poussé l'évolution des conceptions matérielles.

Φ Bus PCI : capacités de transfert de 132 Mo/S ;

Φ Le nouveau bus graphique AGP (Accelerated graphics Port) possède un taux de transfert de 528 Mo/S ;

## 2) Les disques de stockage

Le disque peut être considéré comme le seul périphérique d'E/S commun à tous les ordinateurs. Même les moniteurs et les claviers ne sont pas indispensables sur des systèmes tels que les serveurs. Il existe de nombreuses tailles et vitesses de disques et l'information peut être stockée de manière optique ou magnétique. L'unité élémentaire de stockage d'informations est le secteur.

Φ Les DVD et CD-ROM : les secteurs forment une longue spirale qui s'éloigne en tournant du centre du disque.

Φ Sur les disquettes et disques dans le support tourne à vitesse constante. Les secteurs sont organisés en pistes. Les pistes sont des cercles concentriques autour du centre.

Φ Certains disques stockent le même nombre de secteurs pour chaque piste. D'autres disques placent plus de secteurs sur des pistes externes.

Φ Les disques contiennent un ou plusieurs plateaux de support. Certains proposent d'utiliser les deux faces du plateau et propose deux têtes de lecture/écriture par plateau.

## 3) Ordonnancement du disque dur

Si un disque est appelé à répondre à plusieurs E/S, il doit ordonnancer ces requêtes suivant certains algorithmes d'ordonnancement. La performance de ces algorithmes se mesure par le total des mouvements par tête.

Φ **FIFO (First In, First Out)** : premier entré, premier servi.

Φ **Priorité** : la requête venant du processus ayant la priorité la plus élevée est servi la première.

Φ **SSTF (Shortest Seek Time First)**: traduit plus court positionnement d'abord, répond à la requête dont la position de la piste est la plus proche de celle en cours.

Φ **SCAN** : avance et recule la tête de Lecture/écriture entre la piste la plus interne et la plus externe et satisfait en route toutes les requêtes de la piste en suspens.

Φ **LOOK** : le même que **SCAN**, mais s'il n'y a pas de requêtes dans un sens la tête ne fait pas le déplacement de façon inutile.

- Φ **C-SCAN et C-LOOK** : une fois la dernière piste est atteinte, les algorithmes retournent sur la piste de départ.
- Φ **N-step SCAN** : file de requête divisée en sous files de longueur N. Ces files sont ordonnancées en FIFO. Au sein des files les requêtes sont ordonnancées en SCAN.
- Φ **FSCAN** : tel que le précédent, mais seulement deux sous files illimitées : Une en cours de traitement et l'autre pour les nouvelles requêtes (celles qui viennent lors du traitement).

## III.4. GESTION DES FICHIERS

### III.4.1. Notion

Le volume des données traitées par les applications informatiques atteignant plusieurs méga et giga octets, ces données ne peuvent pas être stockées dans la mémoire centrale. Un stockage persistant à long terme est également nécessaire. Le principe consiste à stocker ces données dans des mémoires secondaires sous forme de fichiers, c'est-à-dire de suites de blocs (la plus petite unité que le périphérique de stockage est capable de gérer). Le contenu de ces blocs, simple suite de données binaires, peut être interprété selon le format de fichier comme des caractères, des nombres entiers ou flottants, des codes d'opérations machines, des adresses mémoires, etc. L'échange entre les deux types de mémoire se fait ensuite par transfert de blocs. L'objectif du système de fichiers est de permettre l'accès au contenu du fichier (l'ouverture du fichier, sa copie dans un second emplacement ou sa suppression) à partir de son chemin d'accès, formé d'un nom précédé d'une liste de répertoires imbriqués.

### III.4.2. Les fichiers

#### 1) Définition

Un **fichier** est une collection logique d'information. Un système de fichiers est une collection de fichiers.

#### 2) Le système de gestion de fichiers

Une des fonctions d'un SE est de masquer les spécificités des disques et des autres périphériques d'E/S et d'offrir au programmeur un modèle de manipulation des fichiers agréable et indépendant du matériel utilisé. Les appels système permettent de créer des fichiers, de les supprimer, de lire et d'écrire dans un fichier. Il faut également ouvrir un fichier avant de l'utiliser, le fermer ultérieurement. Les fichiers sont regroupés en répertoires arborescents; ils sont accessibles en énonçant leur chemin d'accès (chemin d'accès absolu à partir de la racine ou bien chemin d'accès relatif dans le cadre du répertoire de travail courant). Le SE gère également la protection des fichiers.

### III.4.3. Répertoires, noms de fichiers et partitions

#### 1) Répertoire

Les systèmes de fichiers permettent aux utilisateurs d'organiser des fichiers et d'autres objets de systèmes de fichiers au moyen de répertoires. Un répertoire (dossier) est généralement défini comme un objet du système de fichier contenant d'autres objets de systèmes de fichiers. Les entrées des répertoires déterminent le chemin d'accès absolu ou nom associé à un objet du système de fichiers. *En commençant par le répertoire racine, le chemin d'accès absolu est construit en concaténant la séquence des noms parcourus séparé par un «backslash» : \ sous windows ou Dos et un «slash» / sous linux ou Unix.* La plupart des systèmes prennent en charge la notion de répertoire courant ; au lieu de recourir à un chemin d'accès complet, s'avérant très long, il est possible de spécifier un chemin d'accès relatif.

#### 2) Noms de fichiers

Les noms de fichiers représentent des mécanismes d'abstraction qui permettent d'écrire des données et de les trouver plus tard : chaque fichier est référencé par son nom. Le nom d'un fichier possède une longueur maximale qu'il ne peut pas dépasser relativement au système d'exploitation (Dos : 8, Win\_XP : 255). Le nom est généralement composé de la partie nom et la partie extension séparés par un point.

#### 3) Partitions

Les fichiers sont généralement stockés sur des unités de mémoire secondaire ; la mémoire vive peut néanmoins être utilisée pour stocker des fichiers, tels que les fichiers temporaires, pour lesquels un accès rapide est souhaité. La notion de partition peut également entrer en compte pour déterminer sur quelle unité doit être stockée un fichier. Sur certains systèmes, comme Dos et Windows, la partition est spécifiée dans le chemin d'accès. *Le nom c:\rules\II2.txt (sous Dos ou Windows) indique que l'objet du système de fichiers nommé \rules\II2.txt se trouve sur la partition c :*

### III.4.4. Type des objets du système de fichiers

Dans de nombreux systèmes d'exploitation, les noms des systèmes de fichiers peuvent faire référence à des objets qui ne sont ni des fichiers ni des répertoires. Les objets qui peuvent se trouver dans un système de fichiers comprennent les éléments suivants :

Φ **Un raccourci** : un raccourci est un pointeur vers un autre nom dans le système de fichiers. Dans la plupart des cas, le fait de se référer à un raccourci revient à se référer au nom pointé par le raccourci. La suppression du nom vers lequel pointe le raccourci laisse généralement le raccourci en suspens.

Φ **Un périphérique** : un élément matériel, comme un port parallèle.

Φ **Un tube** : un canal de communication entre 2 processus. Un processus envoie des données dans le tube l'autre processus lit ces données depuis le tube. Le tube met en mémoire tampon les données écrites à l'intérieur, jusqu'à ce qu'un autre processus lise les données. Le tampon est généralement de taille limitée et un processus qui écrit sur le tube doit être suspendu lorsque le tampon est plein. Les tubes peuvent être nommés ou non. Comme ils n'ont pas de nom les identifiant, les tubes sans nom ne sont généralement accessibles que par le processus les ayant créés ou les processus qui sont des descendant du processus.

Φ **De la mémoire partagée** : une allocation d'emplacement de mémoire utilisable par un ou plusieurs processus. Comparable à un fichier sur un disque virtuel.

### III.4.5. Fonctions des systèmes de fichiers

Le système de fichier doit offrir aux utilisateurs la possibilité d'accomplir des opérations abstraites sur les objets au sein du système de fichiers. Il doit, au minimum, proposer les fonctions suivantes : créations, suppression, lecture et écriture.

### III.4.6. Architecture du système de fichiers

#### 1) Structure de fichiers

Les fichiers peuvent être structurés de deux manières sous formes de suites d'octets non structurés ou d'une suite d'enregistrements. Un fichier est une séquence d'enregistrements de longueur fixe qui ont la même structure interne. Un fichier prend la forme d'un arbre d'enregistrements qui ne sont pas nécessairement de même longueur. Un enregistrement est une collection logique d'informations (par exemple, une ligne de texte, des informations relatives à une personne). Les opérations d'E/S s'effectuent généralement en termes d'enregistrements. Le système d'exploitation peut gérer des structures d'enregistrements fixes et/ou variables. Les enregistrements peuvent à leur tour être divisés en champs, un champ représentant une donnée élémentaire (telle que le nom et l'âge). La figure ci-dessous décrit la structure logique d'un fichier.

	Chp 1	Chp 2			Chp m
Enregistrement 1	Ali	Salah	1982	...	Brun
Enregistrement 2	Amal	Ali	1986	....	Roux
				.	
				.	
Enregistrement n	Basem	Faker	1910	.....	Blond

Figure n° 3.1.9: Structure logique d'un fichier

Transparent à l'utilisateur, le système d'exploitation considère un fichier comme une collection de blocs logique à taille fixe. Un bloc est l'unité de base d'une opération d'E/S entre le disque et la mémoire tampon du système de fichiers. Le disque en tant que tel est un ensemble de blocs physiques. Chacun d'entre eux stocke un bloc logique et éventuellement d'autres données administratives. La taille du bloc est un multiple de l'unité d'E/S de base fournie par le pilote du disque.

## 2) Méthodes d'accès

Il existe deux méthodes fondamentales pour accéder à des informations au sein d'un fichier : séquentielles et directes. Dans l'accès séquentiel, il faut accéder aux informations du fichier dans l'ordre dans lequel elles ont été stockées dans le fichier. L'accès se déroule de manière séquentielle depuis le début jusqu'à la fin. Les opérations de lecture ou d'écriture sur le fichier n'ont pas besoin de spécifier l'emplacement logique au sein du fichier, car le système d'exploitation maintient un pointeur de fichier qui détermine l'emplacement du prochain accès.

Avec l'accès direct, il est possible d'accéder à tout emplacement logique à l'intérieur du fichier. Généralement, l'accès direct peut être réalisé de 2 manières : Soit en spécifiant l'emplacement logique auquel accéder comme un paramètre à l'opération de lecture ou d'écriture, soit en spécifiant l'emplacement d'une opération de positionnement pour qu'il soit appelé avant la lecture ou l'écriture. Les systèmes de base de données utilisent deux opérations d'accès au système d'exploitation élémentaire pour mettre en œuvre un large éventail de méthodes d'accès de haut niveau. Certains systèmes d'exploitation mettent en œuvre des méthodes d'accès de haut niveau par eux-mêmes. Parmi toutes ces méthodes, l'accès indexé est peut-être le plus significatif. Avec ce dernier, chaque enregistrement de fichiers dispose d'un ou plusieurs champs. Un champ sert de champ d'indexe. Les opérations de lecture et d'écriture comprennent un paramètre d'index. L'enregistrement avec la valeur d'index correspondante est l'enregistrement sur lequel est effectuée l'opération.

Pour toute méthode d'accès, les opérations de lecture et d'écriture peuvent être synchrones ou asynchrones. Les blocs d'E/S synchrones bloquent le processus jusqu'à la fin de l'opération d'E/S. Les E/S asynchrones renvoient immédiatement le contrôle au processus, laissant le processus libre de continuer à s'exécuter pendant l'E/S. Si les opérations d'entrées sont asynchrones, il faut faire appel à certains mécanismes pour avertir le processus que l'opération est terminée. Pour cela, il est possible d'envoyer un signal au processus, d'attribuer une valeur particulière aux variables du processus ou de lancer un appel système spécial pour tester l'état de l'opération d'E/S. Les opérations de sortie asynchrones



peuvent recourir aux mêmes techniques de notifications ou n'offrir aucune notification. L'E/S synchrone représente la norme qui simplifie considérablement la programmation d'application. La grande vitesse des opérations d'E/S de fichier n'incite en rien à l'utilisation des E/S asynchrones. Cependant ces dernières peuvent parfois être employées avec un périphérique d'E/S.

### 3) Contrôle des droits d'accès

Le contrôle des droits d'accès établit une limite quant aux personnes pouvant accéder aux fichiers et à la manière dont elles peuvent y accéder. Le mécanisme de contrôle des droits d'accès le plus simple attribue un accès illimité à tous les utilisateurs. Il s'agit là du modèle de contrôle d'accès choisi par DOS. Sur un tel système, les utilisateurs qui souhaitent contrôler l'accès à leurs fichiers doivent mettre en place une limite d'accès physique (et sur le réseau) de leur machine. Un aspect important du contrôle des droits d'accès est le type d'opérations à réaliser sur le fichier. Les opérations contrôlées comprennent entre autre :

- Φ **La lecture** : lecture des informations contenues dans le fichier.
- Φ **L'écriture** : écriture de nouvelles informations dans un fichier ou écrasement des informations d'un fichier.
- Φ **L'adjonction** : écriture de nouvelles informations à la fin du fichier seulement.
- Φ **La suppression** : suppression d'un fichier et libération de son espace de stockage en vue d'une utilisation dans d'autres fichiers.
- Φ **La liste** : lecture des noms contenus dans un répertoire.
- Φ **L'exécution** : chargement du contenu d'un fichier dans la mémoire principale et création d'un processus pour l'exécuter.
- Φ **Le changement des droits d'accès** : modifications de certains droits d'accès d'utilisateur en vue d'une opération de contrôle.

L'autre caractéristique majeure du contrôle des droits d'accès est la manière dont il détermine ou non d'octroyer l'accès. Le mécanisme le plus commun consiste à baser la décision sur l'identité de l'utilisateur. Sur un système qui utilise une liste des droits d'accès, le système d'exploitation associe à chaque fichier le type d'opérations autorisé à chaque utilisateur. Dans un modèle de droits d'accès illimité, un ensemble indépendant de permissions est conservé pour chaque utilisateur, ce qui représente un volume important de données.

Un mécanisme de contrôle des droits d'accès limité réduit ce volume en regroupant les permissions d'accès pour un certain nombre d'utilisateurs ou de fichiers. Ainsi, de nombreux systèmes d'exploitation mettent en œuvre la notion de groupes d'utilisateurs. Chaque utilisateur et chaque fichier sont associés à un ou plusieurs groupes d'utilisateur. Au lieu d'avoir un ensemble de permissions d'accès pour chaque

utilisateur, le fichier possède uniquement un ensemble de permissions d'accès pour son propriétaire et pour chaque groupe auquel il est associé. Tous les utilisateurs d'un groupe partagent les mêmes permissions d'accès. Un autre groupement fréquemment utilisé consiste à demander à ce que tous les fichiers d'un répertoire partagent les mêmes permissions.

L'autre base communément utilisée pour contrôler l'accès est le mot de passe. Pour réaliser une opération sur un fichier, un utilisateur doit spécifier le mot de passe de fichier associé à cette opération (un mot de passe de fichier est distinct de tout éventuel mot de passe d'ouverture de session). Comme avec les listes de droits d'accès, le groupement peut réduire le volume des données maintenues par le système d'exploitation (et réduire le nombre de mots de passe que doit retenir un utilisateur). Ainsi, tous les fichiers d'un répertoire peuvent partager le même mot de passe. Les capacités constituent une variante intéressante des listes de droits d'accès, mais cependant moins répandue. Sur les systèmes basés sur les capacités, les droits d'accès, au lieu d'être associés aux fichiers, sont associés aux processus. Lorsqu'un accès au fichier est tenté, le système d'exploitation vérifie le droit correspondant au fichier dans les droits d'accès associés au processus.

#### 4) Verrouillage des fichiers

Le verrouillage des fichiers offre aux processus la possibilité de mettre en œuvre un accès exclusif à un fichier. Trois grandes options existent dans la mise en œuvre du verrouillage :

- Le verrouillage peut se limiter à l'ensemble des fichiers ou la mise en œuvre peut autoriser de verrouiller certaines parties d'un fichier.
- Le verrouillage peut s'appliquer à tout accès ou il peut exister différents niveaux. Sur certains systèmes, il y a la fois des blocs de lecture et d'écriture. Un fichier verrouillé pour la lecture n'empêche pas un autre accès en lecture, mais l'accès en écriture par d'autre processus est refusé.
- Le verrouillage peut être soit obligatoire soit consultatif. Avec le verrouillage obligatoire, le système d'exploitation refuse l'accès pendant que le fichier est verrouillé. Avec le verrouillage consultatif, les primitives de verrouillage fournissent uniquement des informations sur l'état de verrouillage du fichier. L'accès n'est restreint que si un processus vérifie l'état de verrouillage du fichier et respecte le verrouillage qui est indiqué. Dans certaines circonstances, un système d'exploitation peut mettre en œuvre un mécanisme de verrouillage implicite.

#### 5) Attribution des blocs

La méthode d'attribution de blocs détermine comment les enregistrements d'un fichier sont attribués dans des blocs.

- ∅ **Attribution fixe des blocs** : pour les fichiers avec des enregistrements de taille fixe, un nombre intégral d'enregistrements est stocké dans chaque bloc. Aucun enregistrement ne peut être plus grand qu'un bloc. Si la taille du bloc n'est pas un multiple de la taille de l'enregistrement, il y aura de l'espace inoccupé à la fin du bloc. Le système d'exploitation peut calculer le bloc et l'offset à l'intérieur du bloc de tout enregistrement, en fonction de la taille de l'enregistrement et du bloc.
- ∅ **Attribution non étendue de blocs** : pour les systèmes avec des enregistrements de taille variable, plusieurs enregistrements peuvent être stockés dans chaque bloc, mais aucun ne peut s'étendre sur plusieurs blocs. Les enregistrements ne peuvent pas non plus être plus grands que la taille du bloc. L'espace à la fin d'un bloc est gaspillé si le prochain enregistrement est plus important que cet espace. Le système d'exploitation ne peut calculer l'emplacement d'un enregistrement, à moins qu'il ne connaisse la taille de tous les enregistrements qui le précède.
- ∅ **Attribution étendue de blocs** : les enregistrements peuvent être stockés dans plusieurs blocs. Il n'existe aucune limite quant à la taille d'un enregistrement et il n'y a pas d'espace inutilisé à l'intérieur d'un bloc. La seule manière de calculer l'emplacement d'un enregistrement d'un enregistrement consiste à additionner la taille de tous les enregistrements qui le précède.

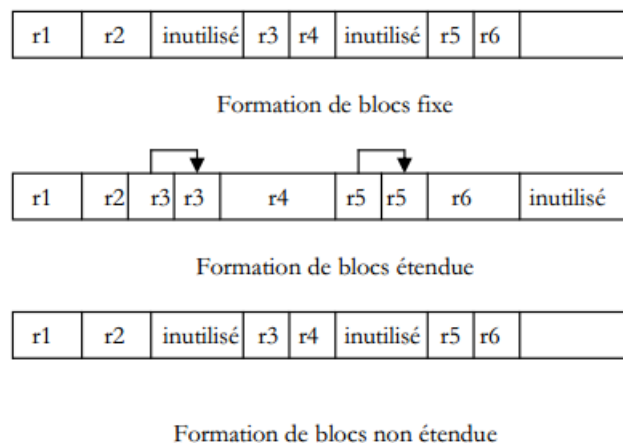


Figure n° 3.1.10: Architectures des formations des différents blocs

### III.4.7. Allocation

Il existe 3 modèles de base pour allouer l'espace de la mémoire auxiliaire à des fichiers. Le schéma d'allocation est chargé d'associer des blocs logiques d'un fichier à des blocs physiques de la mémoire auxiliaire. Dans la plupart des systèmes d'exploitation, la taille d'un bloc physique est une puissance de 2 comprise entre 512 et 4096.

### 1) Allocation contiguë

Le modèle le plus simple est l'allocation contiguë. Les blocs logiques d'un fichier sont stockés dans une partition de blocs physiques contigus. L'entrée du répertoire a uniquement besoin de stocker l'adresse de la mémoire auxiliaire de départ du fichier ainsi que la taille de ce dernier. L'emplacement physique de tout octet du fichier peut être calculé en ajoutant l'offset approprié à l'adresse de la mémoire auxiliaire de départ de fichier.

Lorsqu'un fichier est créé, l'allocation contiguë requiert une pré-allocation d'espace pour le fichier. Le système d'exploitation peut être conçu pour étendre l'allocation du fichier, si nécessaire. Si l'expansion est autorisée et que l'espace de stockage au-dehors de la fin du fichier est utilisé, un ou plusieurs fichiers doivent être déplacés pour loger le fichier le plus important. Les fichiers à déplacer doivent recevoir de nouvelles partitions sur le disque, puis être copiés sur ces nouvelles partitions.

### 2) Allocation chaînée

Dans l'allocation chaînée, les blocs physiques dans lesquels est stocké un fichier peuvent être dispersés dans l'ensemble de la mémoire auxiliaire. Les blocs physiques sont plus importants que les blocs logiques et stockent à la fois le bloc logique et un pointeur vers le bloc physique dans lequel est stocké le prochain bloc logique du fichier. L'entrée du répertoire stocke l'emplacement du premier bloc physique. Le bloc physique associé au N<sup>ème</sup> bloc logique peut être déterminé uniquement en lisant les précédents blocs N-1 et en suivant les liens qu'ils contiennent. Les performances des opérations d'adjonctions (écriture à la fin d'un fichier) peuvent être améliorées de façon significative en incluant également dans l'entrée du répertoire un pointeur vers le dernier bloc de la file.

### 3) Allocation indexée

L'allocation indexée est une variante de l'allocation chaînée. Le bloc physique stocke seulement le bloc logique, qui est par conséquent de la même taille qu'un bloc logique. Les liens vers les blocs physiques d'un fichier sont stockés de manière contiguë dans une table d'index. L'entrée du répertoire contient soit la table d'index soit un pointeur vers celle-ci.

Avec les allocations contiguës et chaînées, il suffit au système de fichiers de stocker l'emplacement physique de départ du fichier. Toutes les autres adresses peuvent être déterminées à partir de l'emplacement de départ du fichier. Avec l'allocation indexée, le système de fichiers doit avoir une entrée d'index pour chaque bloc du fichier. Pour minimiser le volume d'espace requis dans les structures de répertoire, l'indexation à plusieurs niveaux peut être utilisée.

#### 4) Espace libre

Outre le fait de garder la trace de l'emplacement de chaque fichier dans la mémoire auxiliaire, un système d'exploitation doit aussi être capable d'identifier les blocs physiques qui ne sont alloués à aucun fichier. Un certain nombre de mécanismes ont été utilisés pour maintenir la liste d'espace libre. Un bloc est soit utilisé soit non utilisé ; son statut peut donc être stocké dans un seul bit. L'utilisation d'une série de bits, un bit par bloc physique, crée un champ de bits d'espace libre. Les bits sont regroupés en aussi peu de mots que possible. Bien que ce mécanisme fasse un usage efficace de l'espace, l'accès aux informations sur un bloc particulier est ralenti par le traitement supplémentaire nécessaire à l'extraction des informations en bits individuels depuis un mot.

Un tableau d'espace libre stocke les adresses de tous les blocs libres. Un bloc est utilisé si son adresse n'est pas dans le tableau. Un tableau permet une implémentation efficace d'opérations d'ajout ou de suppression des blocs de la liste d'espace libre. Cette liste peut également être maintenue en créant une liste chaînée de tous les blocs libres. L'avantage de ce schéma est l'efficacité de l'utilisation de l'espace ; les blocs libres stockent eux-mêmes des informations relatives à la liste d'espace libre.

Les logiciels de gestion des ressources informatiques impliquent *le processus d'identification, d'acquisition, de mise en service, de maintenance et d'élimination des périphériques et applications de réseau.*

- ☑ **Identification** : Les services informatiques doivent identifier le type de périphériques et de logiciels requis pour les différents services de leur entreprise. Une fois ces types identifiés, l'étape suivante consiste à chercher des fournisseurs à sélectionner et à déterminer la durée pendant laquelle vous souhaitez acheter ces logiciels sous licence.
- ☑ **Acquisition** : Les ressources peuvent être acquises sous différentes formes ; par exemple, elles peuvent être développées, achetées ou prêtées.
- ☑ **Intégration** : Après l'achat, la ressource doit être installée à l'intérieur du réseau et intégrée à d'autres éléments essentiels de votre entreprise.
- ☑ **Maintenance** : Un bon entretien de ces ressources permettra de réduire les frais généraux de votre entreprise. Un entretien régulier, des réparations et des mises à jour joueront un rôle essentiel dans le cycle de vie.
- ☑ **Élimination** : Une fois que les ressources ont dépassé leur durée de vie, les services informatiques doivent les céder et en acquérir de nouvelles en fonction de ce qui est disponible sur le marché actuel ainsi que des besoins de l'entreprise au moment de l'achat.

## Conclusion

Relativement, l'ensemble de tout ce qui précède n'est pas de comprendre comment fonctionne un OS, cependant la prise en main d'un système d'exploitation, selon la prédilection de l'enseignant. Avant de basculer dans cette phase pilote nous avons d'abord présenté quelques aspects de la logique de tel ou tel autre OS, qui justifie bon nombre d'astuces difficiles à cataloguer autrement. Ainsi, la connaissance des parties précédentes permettra d'exploiter aux mieux tel ou tel autre système. Nous avons également évoqué la plupart des tâches courantes et la façon de s'en acquitter, ce qui doit permettre aux nouveaux venus de retrouver rapidement leurs marques d'usages d'un Système d'exploitation.

De son côté, l'informaticien doit être capable de déterminer ce qu'il veut faire, protéger et contre qui. Connaissant ses propres vulnérabilités, une analyse de risque lui permettra d'identifier les scénarios d'attaques réalistes et par conséquent de mettre en place les parades nécessaires à la protection de l'OS de son choix. Bien plus, avec beaucoup d'aspects techniques et une forte compétition industrielle. Aujourd'hui, il existe un système d'exploitation dans chaque ordinateur, tablette, téléphone portable, et plus généralement tout objet numérique, avec sur le marché des dizaines de systèmes d'exploitation différents. Dans leur nom, on trouve souvent le sigle OS pour Operating System, en anglais. Même si les constructeurs utilisent des noms différents pour se démarquer de leurs concurrents, on retrouve dans tous les systèmes d'exploitation des aspects communs et des invariants.

De surcroît, d'un ordinateur ou d'une tablette, on voit d'abord l'écran et le boîtier qui enferme le matériel informatique (processeurs, mémoires et canaux d'entrées-sorties), toutefois, le système d'exploitation vient alors au secours de l'utilisateur, en lui permettant d'exploiter l'ordinateur mieux qu'en langage machine. Mais il fait plus que cela. Qui plus est, le but d'un système d'exploitation est de rendre aisée l'utilisation de l'ordinateur par chacun, comme s'il s'agissait d'une machine fictive, sa « machine virtuelle », qui aurait été construite pour lui. Le système fournit un accès commode et ergonomique.

Ni, le système d'exploitation est la passerelle entre l'utilisateur, les ressources et les applications. Lorsqu'un programme est lancé, il ne communique pas directement avec un périphérique. Les instructions passent par le BIOS qui fonctionne en mode arrière-plan puis le système d'exploitation, qui se charge de les transmettre au périphérique. Les différents systèmes disponibles sur le marché se différencient de plusieurs façons, notamment par leur caractère multitâche (capacité à exécuter plusieurs processus en même temps) et par la longueur des informations qu'ils sont capables de gérer en même temps. Cette longueur est exprimée en bits (16, 32 à 64 bits en général, plus chez les systèmes très performants).

En tout état de cause, l'OS joue aussi un rôle clé dans la gestion des ressources. Il dissocie les ressources matérielles et les programmes, afin de mettre à disposition de l'utilisateur une interface plus simple, plus intuitive et plus facile à manier. Cette gestion se décline en plusieurs volets. A l'aide d'un algorithme, le système d'exploitation gère l'allocation du processeur. Il intervient aussi dans la gestion de la mémoire vive, qui est limitée. Lorsqu'un manque d'espace libre est constaté, il crée des espaces libres appelés mémoire virtuelle. L'OS agit également sur le lancement et l'exécution des applications, en s'assurant que tous les ingrédients nécessaires à leur fonctionnement soient réunis et en éliminant celles qui présentent un dysfonctionnement. Son rôle s'étend à la gestion des fichiers (lecture, écriture et accès), des informations, des entrées et des sorties. Les systèmes d'exploitation les plus utilisés sont Windows (3.1 / 95 / 98 / 2000 / XP / 7 / 10), Mac OS X et Linux.

Enfin, le système d'exploitation est un ensemble de logiciels composé d'un noyau, le central de ce complexe qui réalise toutes les tâches qui lui sont attribuées. La communication entre l'utilisateur et l'OS est assurée par l'interpréteur de commande, parfois surnommé « coquille » (shell en anglais). Il renferme également un système de fichiers. Les systèmes d'exploitation multitâches présentent un grand intérêt pour les professionnels, notamment dans le cadre du travail collaboratif. Plusieurs utilisateurs peuvent utiliser la même machine sans avoir l'impression que d'autres personnes sont connectées et peuvent, elles aussi, réaliser des actions. Pour jouer son rôle de passerelle entre l'utilisateur et les applications, l'OS a employé des pilotes matériels.