

# Chapitre 1 :

## Introduction aux Systèmes d'Exploitation

### 1. Introduction :

Les ordinateurs permettent de collecter des données, de réaliser des calculs, de stocker des informations et de communiquer avec d'autres ordinateurs.

Un ordinateur est formé d'une partie matérielle et d'une partie logicielle. Cette dernière comporte des logiciels qui sont classés en deux catégories : les programmes d'application des utilisateurs et les programmes système qui permettent le fonctionnement de l'ordinateur. Parmi ceux-ci, le système d'exploitation (SE).

### 2. Définition :

- Le système d'exploitation est un programme qui gère et contrôle les composants de l'ordinateur. Il fournit une base appelée machine virtuelle, sur laquelle seront construits les programmes d'application et les utilitaires au moyen des services ou appels système.
- Le S.E est un intermédiaire entre l'utilisateur (programme d'application) et la machine physique.

### 3. Role :

Le rôle du S.E est de résoudre les problèmes relatifs à l'exploitation de l'ordinateur en garantissant :

- Une gestion *efficace, fiable et économique* des ressources physiques de l'ordinateur (notamment les ressources critiques telles que processeur, mémoire...) : il ordonne et contrôle l'allocation des processeurs, des mémoires, des icônes et fenêtres, des périphériques, des réseaux entre les programmes qui les utilisent. Il *assiste* les programmes utilisateurs. Il *protège* les utilisateurs dans le cas d'usage partagé.
- Il propose à l'utilisateur une *abstraction* plus simple et plus agréable que le matériel : une *machine virtuelle* permettant l'interaction avec les utilisateurs en leur présentant une machine plus simple à exploiter que la machine réelle.

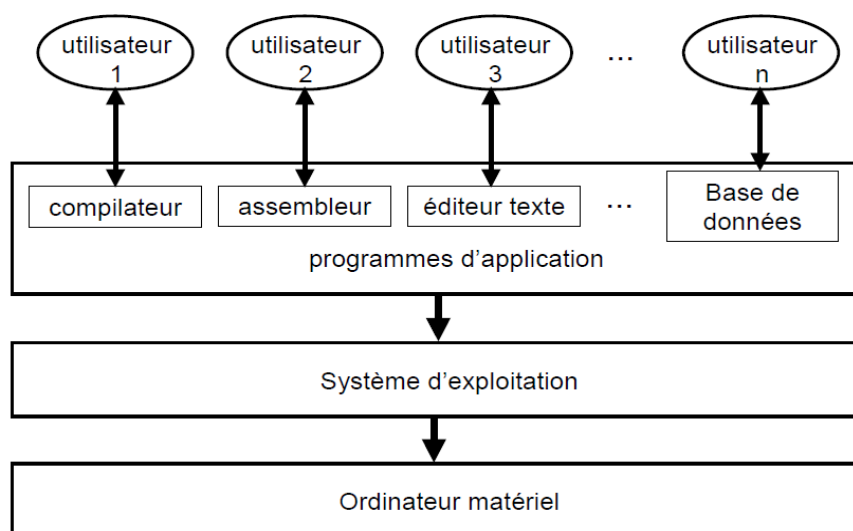


Fig1. Les couches d'un système informatique

### 4. Historique des S.E :

L'historique des S.Es peut être répartie en plusieurs génération :

#### 4.1. Génération de la « Porte Ouverte » 1945-1955 :

- Construction des machine avec les tubes électroniques.
- Machines énormes, coûteuses.
- Absence de S.E.
- Écriture des programmes en langage machine.
- réservation de la machine pour une tranche de temps pour chaque programmeur (un seul programme en mémoire).
- Introduction des cartes perforées en 1950.

#### 4.2. Génération de « Traitement par Lots » 1955-1965

- Construction des machines avec les transistors et l'utilisation des bandes magnétiques.
- Écriture des programmes en Fortran ou en assembleur sur des cartes perforées.
- Le mode de fonctionnement est le traitement par Lots : Transférer les travaux sur une bande magnétique, Monter la bande sur le lecteur de bandes, Charger en mémoire un programme spécial (l'ancêtre des systèmes d'exploitation) qui lit puis exécute, l'un à la suite de l'autre, les programmes de la bande. Les résultats sont récupérés sur une autre bande, à la fin de l'exécution de tout le lot, Imprimer les résultats.

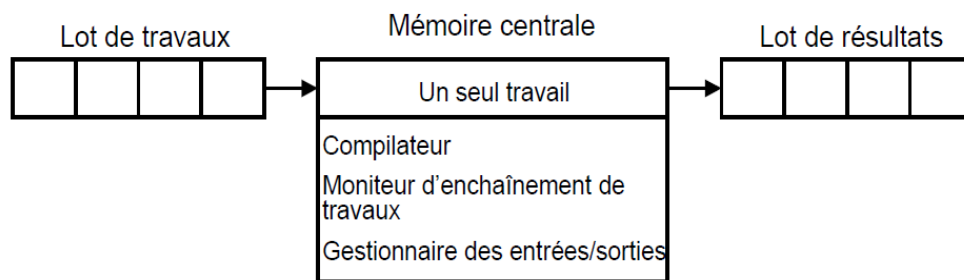


Fig2. Traitement par Lots

- Le processeur restait inutilisé pendant les opérations d'entrée et sortie (E/S).

#### 4.3. Génération de « Multiprogrammation & Circuits Intégrés & Temps partagé » 1965-1980 :

- Circuits intégrés : leur avantage au plan prix/performance a permis la création d'une seule gamme de produits avec la même architecture matérielle et le même jeu d'instructions et un même programme qui peut tourner sur toutes ces machines (pas de machine d'E/S et d'autres pour le calcul).

- Nouvelles techniques dans les systèmes d'exploitation :

✓ Multiprogrammation : charger plusieurs programmes dans la MC et de les exécuter en fonction des ressources qu'ils utilisent. Elle vise l'élimination des temps d'attente de l'UC pendant les opérations d'E/S. Entre autre, ce mécanisme de multiprogrammation a exigé l'apparition de la Mémoire partagée et les mécanismes de protection.

✓ Spool (Simultaneous Peripheral Operation On Line): transfert des travaux des cartes vers le disque permettant la lecture et l'écriture simultanées sur le disque et disparition de la manipulation des bandes.

✓ Partage de temps : une variante de la multiprogrammation (chaque utilisateur possède un terminal en ligne) qui permet de partager le temps de l'UC entre les programmes qui résident dans la MC

- Exemples de systèmes d'exploitation :
  - MULTICS (MULTiplexed Information and Computing Service): système pour ordinateur central, capable de supporter des centaines d'utilisateurs simultanés .
  - DEC et PDP-1, ... PDP-11 : Systèmes pour mini-ordinateurs ; performant pour des tâches non scientifiques ; de 4K mots de 18 bits, pour un prix de 120 000 \$).
  - UNICS (Uniplexed Information and Computer Service) : une version simplifiée (mono-utilisateur) de MULTICS qui allait devenir le célèbre UNIX écrit en langage C et le plus porté sur toutes sortes de machines.

#### 4.4. Génération des ordinateurs personnels (1980-aujourd'hui) :

• Construction des machines avec les Circuits LSI (Large Scale Integration) : Circuits intégrés à haute densité contenant des milliers de transistors sur 1mm<sup>2</sup> de silicium permettant le développement des micro-ordinateurs (ordinateurs personnels), très peu onéreux comparés aux mini-ordinateurs de type PDP-11

• Exemples de systèmes d'exploitation

- ✓ CP/M (Control Program for Microcomputers) : conçu pour Intel 8080, comprend un contrôleur pour le tout récent lecteur de disquettes 8 pouces .
- ✓ MS-DOS (MicroSoft-Disk Operating System): développé en 1980 pour l'IBM PC par B.Gates à partir du noyau DOS. Il intègre petit à petit des concepts riches d'UNIX et de MULTICS
- ✓ Macintosh d'Apple : un S.E qui intègre une interface graphique, destiné à des utilisateurs qui ne connaissaient rien aux ordinateurs.
- ✓ Windows de Microsoft : successeur de MS-DOS qui intègre une interface graphique influencé par le succès de Macintosh (Windows 95/98/NT/2000/XP).
- ✓ UNIX ; Linux : intègre de l'interface graphique avec un système de fenêtres appelé X-Window.

### **5. Fonctions & Structure en couches d'un SE :**

Un S.E peut être structuré sous forme de plusieurs couches :

- **Le noyau** réside en mémoire et est de petite taille . Il assure les fonctions suivantes :
  - Gestion du processeur: responsable de la répartition du temps processeur entre les différents processus.
  - Gestion des interruptions : les interruptions sont des signaux envoyés par le matériel, à destination du logiciel, pour signaler un événement.
  - Gestion du multi- tâches: simuler la simultanéité des processus coopératifs (i. e. les processus devant se synchroniser pour échanger des données) et gérer les accès concurrents aux ressources (fichiers, imprimantes, ...)
- **le gestionnaire de la mémoire** qui se charge du partage de la mémoire entre les processus en attente d'exécution.
- **Le gestionnaire des entrées/sorties** qui se charge de gérer tous les périphériques (clavier, écran, disques, imprimantes, etc.).
- **Le gestionnaire de fichiers** qui se charge de la gestion de l'espace du disque, de la manipulation des fichiers tout en assurant l'intégrité des données, la protection des fichiers, etc.
- **Le module d'allocation de ressources** qui se charge d'assurer une bonne utilisation des ressources ; de comptabiliser et de fournir des statistiques sur l'exploitation des ressources principales ; de permettre à chaque processus existant dans le système d'obtenir les ressources nécessaires dans des limites de temps raisonnables ; d'exclure mutuellement les processus qui demandent une ressource non partageable et d'éviter les situations de blocage.

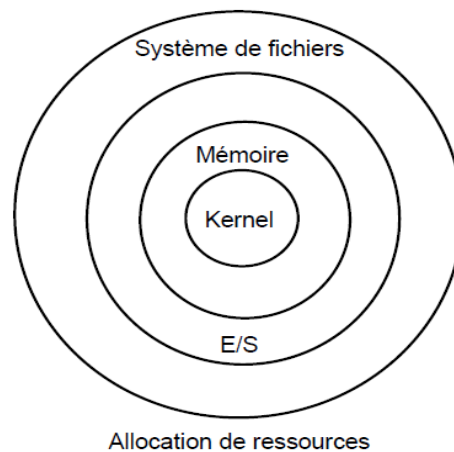


Fig4. Structure en couche d'un S.E

## 6. Interfaces d'un S.E :

Un S.E offrent généralement aux utilisateurs deux interfaces :

- **Interface “programmatique”, ou API (Application Programming Interface) :**  
Pour un utilisateur, le système d’exploitation apparaît comme un ensemble de procédures, trop complexes pour qu’il les écrive lui-même. Les bibliothèques des appels système sont alors des procédures mises à la disposition des programmeurs. Ainsi un programme C/C++ peut utiliser des appels système d’Unix/Linux comme `open()`, `write()` et `read()` pour effectuer des Entrées/Sorties de bas niveau.
- **Interface de l'utilisateur, ou interface de commande :**  
C'est une interface utilisée par un usager humain sous forme textuelle et/ou graphique. L'interpréteur de commandes constitue la forme textuelle. Il est disponible dans tous les systèmes. Il est lancé dès la connexion au système et invite l'utilisateur à introduire une commande. L'interpréteur de commandes récupère puis exécute la commande par combinaison d'appels système et d'outils (compilateurs, éditeurs de lien, etc.). Il affiche les résultats ou les erreurs, puis se met en attente de la commande suivante. Par exemple, la commande de l'interpréteur (shell) d'Unix suivante permet d'afficher à l'écran le contenu du fichier *archive*: ***cat archive***.  
La forme graphique est disponible dans les S.Es offrant un système graphique de fenêtrage tel que Windows, Linux, Macintosh.  
Par exemple si un utilisateur souhaite copier le fichier *fich1* dans le répertoire *rep1*, il peut une des solution suivante :
  - utiliser la forme textuelle en lançant l'interpréteur de commande et d'exécuter la commande *cp fich1 rep1*.
  - utiliser la forme graphique en sélectionnant l'icône du fichier *fich1* par la souris et la déplaçant vers l'icône du repertoire *rep1*.

## 7. Types de S.E :

On distingue plusieurs types de systèmes d'exploitation, selon les services rendus, leur architecture, leur capacité à évoluer et leur architecture matérielle qui les supporte

### 7.1. Selon les services rendus

#### 7.1.1 Systèmes mono/multitâches

- 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 si la précédente est achevée, si non le système se bloque. Cas de MS-DOS.

- Multi-tâches : capacité du système à pouvoir exécuter plusieurs tâches « processus » simultanément ; par exemple effectuer une compilation et consulter le fichier source du programme correspondant. Cas d'UNIX et Windows 95 .

#### **7.1.2 Système mono/multi-utilisateurs :**

- Mono-utilisateur : le système ne peut gérer qu'un seul utilisateur (ex : le système MS-DOS).  
- Multi-utilisateurs : capacité à pouvoir gérer un panel d'utilisateurs utilisant simultanément les mêmes ressources matérielles.

### **7.2. Selon leur architecture**

#### **7.2.1 Systèmes centralisés :**

L'ensemble du système est entièrement présent sur la machine considérée. Les machines éventuellement reliées sont vues comme des entités étrangères disposant elles aussi d'un système centralisé. Le système ne gère que les ressources de la machine sur laquelle il est présent.

Exemple : UNIX, même si les applications réseaux (X11, FTP, Mail ...) se sont développées.

#### **7.2.2 Systèmes répartis (distributed systems) :**

Les différentes abstractions du système sont réparties sur un ensemble (domaine) de machines (site). Le système d'exploitation réparti apparaît aux yeux de ses utilisateurs comme une machine virtuelle, monoprocesseur même lorsque cela n'est pas le cas. Avec un système réparti, l'utilisateur n'a pas à se soucier de la localisation des ressources. Quand il lance un programme, il n'a pas à connaître le nom de la machine qui l'exécutera. Ces systèmes exploitent au mieux les capacités de parallélisme d'un domaine et ils offrent des solutions aux problèmes de la résistance aux pannes.

### **7.3. Selon leur capacité à évoluer**

#### **7.3.1 Systèmes fermés (ou propriétaires) :**

Système avec extensibilité réduite : Quand on veut rajouter des fonctionnalités à un système fermé, il faut remettre en cause sa conception et refaire une archive (système complet). Cas d'UNIX et MS-DOS.

Il n'y a aucun ou peu d'échange possible avec d'autres systèmes de type différent, voir même avec des types identiques.

#### **7.3.2 Systèmes ouverts :**

Système avec extensibilité accrue : Il est possible de rajouter des fonctionnalités et des abstractions sans avoir à repenser le système et même sans avoir à l'arrêter sur une machine.

Cela implique souvent une conception modulaire basée sur le modèle « client-serveur » et la communication entre systèmes, nécessitant des modules spécialisés.

### **7.4. Selon l'architecture matérielle qui les supporte**

#### **7.4.1 Architecture monoprocesseur (temps partagé ou multiprogrammation) :**

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.

#### **7.4.2 Systèmes multiprocesseurs**

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). On trouve une grande variété d'architectures multiprocesseurs :

- ✓ SIMD (Single Instruction Multiple Data) : Tous les processeurs exécutent les mêmes instructions mais sur des données différentes.
- ✓ MIMD (Multiple Instructions Multiple Data) : Chaque processeur est complètement indépendant des autres et exécute des instructions sur des données différentes.
- ✓ Pipeline : Les différentes unités d'exécution sont mises en chaîne et font chacune partie du traitement à effectuer.

On appelle SMP (Symmetric Multiprocessing ou Symmetric Multiprocessor) une architecture dans

laquelle tous les processeurs accèdent à un espace mémoire partagé.

Un système multiprocesseur doit donc être capable de gérer le partage de la mémoire entre plusieurs processeurs mais également de distribuer la charge de travail.

**7.6. Selon 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.

**7.7. Cas particulier : systèmes embarqués, systèmes temps réel**

#### **7.7.1 Systèmes embarqués**

Les 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.

Les principaux systèmes embarqués «grand public» pour assistants numériques personnels sont : Windows CE / Windows Mobile / Window Smartphone .

#### **7.7.2 Systèmes temps réel ( RTOS)**

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.

Exemples de systèmes d'exploitation temps réel : OS-9 ; RTLinux (RealTime Linux) ; QNX ; VxWorks.