

Introduction Aux Systemes d'Exploitation

I. Definition

Un systeme d'exploitation (SE) est la partie la plus importante d'un systeme informatique. Il est constitue d'un ensemble de programmes agissant comme interface entre l'utilisateur et le materiel. Son but principal est de rendre le systeme informatique plus pratique a utiliser d'une maniere efficace et optimale.

L'ensemble materiel et logiciels dans une architecture a processeur constitue ce qu'on appelle un Systeme Informatique (SIQ) auxquels on rajoute les programmes d'applications et les utilisateurs (Figure I 1).

⇒ **Le materiel** : toute architecture a microprocesseur regroupant une UC, une MC, des peripheriques d'E/S, ... et constituant l'ensemble des ressources de base du SIQ.

⇒ **Les programmes d'applications** regroupent les utilitaires aidant a la mise au point des programmes (Assembleurs, compilateurs, editeurs de liens, ...) ainsi que les SGBD, les applications PAO, bureautiques, ...

Ces applications consomment des ressources pour pouvoir resoudre le probleme vise.

⇒ **L'utilisateur** inclut toute entite demandant un service de la machine, que ce soit un etre humain, un ordinateur, un capteur ou une architecture a microprocesseur (robot, machine FAO, surveillance, ...),... etc.

- **Ensemble de Programmes** : un SE est un environnement constitue de plusieurs modules dont chacun a une fonction bien determinee (editeurs de liens, chargeurs, scheduler, protection et securite, gestion des ressources, ...). Mais ceci n'empêche plus qu'une partie d'un SE soit implémentée physiquement ou câblée (la table des pages, l'instruction TAS (ou XCHG), ...)
- **Utilisation Pratique** : la machine est censée être utilisée par des utilisateurs non forcément spécialistes, voire même des néophytes. Cependant l'interaction directe avec elle nécessite une connaissance des moindres détails du matériel ce qui rend l'utilisation du matériel difficile et fastidieuse. Le SE s'interface entre l'user et la machine pour fournir un environnement plus commode à l'user et le détacher au maximum possible de toute connaissance technique du matériel en fournissant une interface Homme Machine plus intuitive, plus conviviale et plus pratique (principe de la machine virtuelle).
- **Utilisation efficace et optimale** : le SE doit contrôler le comportement des différentes applications des utilisateurs et celui des periph d'E/S pour garder le SIQ dans un état cohérent en évitant les erreurs et les utilisations incorrectes de la machine par ces programmes. De plus, il doit appliquer une stratégie d'allocation des ressources (matérielles ou logicielles) rigoureuse pour en optimiser l'exploitation

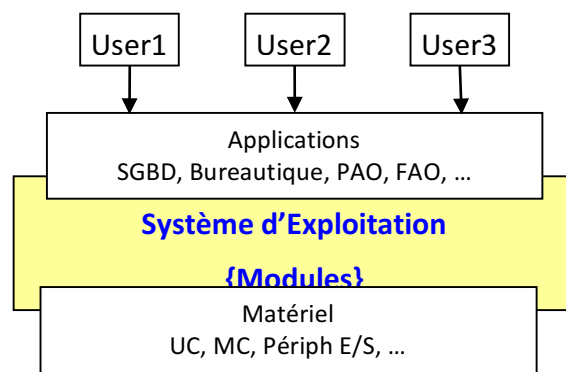


Figure I-1 : Les composants d'un Systeme Informatique

(Les performances du système). Il doit appliquer également une stratégie de sécurité gardant le SIQ loin de tout accès non autorisé (Mode d'exécution, groupes d'utilisateurs, cryptage de données, ...)

## II. Historique

Les SE sont largement liés aux éléments matériels formant l'architecture physique des machines. L'évolution de l'un des deux a été largement condamnée par l'évolution de l'autre. Ils se développent d'une manière interactive. On peut distinguer quatre générations bien différentes :

### 1. Les machines dénudées (1946 – 1955) : cette génération est caractérisée principalement par :

- Machine dédiée : conçue pour un travail bien spécifique.
- Absence de mémoire
- Absence de système d'exploitation, l'utilisateur interagit directement avec la machine :

⇒ Chargement (En binaire en utilisant des interrupteurs puis en hexa en utilisant un clavier hexadécimal !) et débogage manuels des programmes.

⇒ Les E/S sont gérées par l'utilisateur : ce dernier est censé connaître les moindres détails techniques de chaque périphérique.

⊗ *Problème : utilisation lourde et fastidieuse de la machine limitée aux techniciens.*

➔ *Solution : Décharger l'utilisateur au maximum possible des détails techniques de la machine et automatiser les opérations manuelles.*

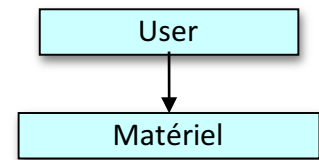


Figure II-1 : Machine dénudée

### Deux événements principaux ont caractérisé la fin de cette génération

- La Conception de quelques utilitaires systèmes (assembleurs, éditeurs de liens, chargeurs) et l'apparition des compilateurs et les langages évolués (Fortran). Ceci a permis de créer des bibliothèques de fonctions réutilisables.
- L'apparition de quelques périph d'E/S (lecteurs de cartes perforées, des bandes magnétiques et des imprimantes) auxquels on a écrit, une fois pour toutes, les drivers avec lesquels les programmes utilisateurs vont s'interagir.

### 2. SE de traitement par lots (batch) (1955 – 1965)

- L'apparition du transistor a permis la production et la commercialisation des machines.
- Utilisation intense des langages évolués et les assembleurs.
- Les différentes étapes de l'exécution d'un programme sont :
  1. L'écriture du code source, en ASM ou en Fortran généralement, qui sera ensuite gravé sur des cartes.
  2. L'opérateur charge les cartes du compilateur adéquat qui lit le programme source (qui est aussi sur des cartes) comme données et génère un code assembleur.
  3. L'opérateur charge les cartes de l'assembleur qui aura comme donnée les cartes générées par le compilateur et génère, après une édition de lien, un exécutable qui peut être chargé pour être exécuté.

⊗ *Problème : les chargements manuels successifs ralentissent le rythme des exécutions. Ce qui donne un rendement de l'UC très faible.*

➔ *Solution : Accélération des chargements en les automatisant.*

Pour accélérer le rythme des chargements, deux méthodes sont introduites :

- a) Le traitement par lots (batch) : consiste à regrouper les travaux similaires et les traiter à la suite. Ainsi tous les travaux Fortran sont rassemblés dans un lot. Ceux de l'assembleur dans un deuxième et ceux de Cobol dans un troisième, etc...

- b) Le moniteur d'enchaînement : l'acheminement automatique des étapes précédentes est assuré par un programme spécial résident en mémoire centrale appelé le moniteur d'enchaînement (ME). La notion de job est apparue.
- Un **job** est traitement utilisateur constitué d'un ensemble de pgmes, de données et des directives qui décrivent l'enchaînement de l'exécution des différentes étapes des pgmes (compilation, exécution, E/S, ...).

### Remarques

- ⇒ Les systèmes batch sont très appropriés à l'exécution des travaux longs qui ont besoin de peu d'interaction avec l'utilisateur.
- ⇒ Le ME est considéré comme étant l'ancêtre des SE. Il réside dans la MC dans une partie privilégiée inaccessible aux utilisateurs.
- ⇒ 1969 : apparition de Unix réécrit en 1973 en langage C.

### ⊗ Problèmes

1. Dans toutes les générations précédentes c'est l'UC qui est chargée d'effectuer les opérations d'E/S.
2. Les périphériques d'E/S sont généralement très lents par rapport à l'UC, ce qui condamne largement son efficacité

### 3. Les systèmes multiprogrammés (1965 – 1980)

- L'apparition des circuits intégrés a favorisé la commercialisation à grande échelle des machines. L'absence de normes standardisées a donné naissance au problème de compatibilité.
- L'apparition des disques durs ayant des capacités de stockage puissante et une rapidité de transfert plus grande a comblé l'inconvénient des bandes magnétiques et a donné naissance à la multiprogrammation.

#### 3.1. La Multiprogrammation : Recouvrement d'E/S

- Les périphs d'E/S sont généralement très lents par rapport à l'UC ; cette dernière est condamnée à suivre leur rythme lorsqu'elle exécute une opération d'E/S donnant lieu à un temps perdu où l'UC chôme (l'ordre de 1e seconde sur 11,5 jours pour une frappe d'une touche sur 0.1s et la prise en compte de cette touche en une 10E-6 secondes) !
- L'idée de donner une autonomie aux périphériques d'E/S en les dédiant d'un petit processeur de gestion des opérations d'E/S, appelé DMA, a permis le recouvrement de phases de calcul d'un programme avec des phases d'E/S d'un autre programme, ces dernières étaient des temps perdus dans les système Batch, ce qui a donné naissance au concept de la multiprogrammation.
- Plusieurs programmes sont chargés sur le disque dur et en MC. L'UC commence l'exécution de l'un des programmes chargés en MC. Dès que celui-ci effectue une opération d'E/S, qui est largement lente par rapport à l'UC, l'UC la lance en initialisant le DMA et entame l'exécution d'un autre programme, sans attendre l'achèvement de l'opération l'E/S. La fin de cette dernière est signalée à l'UC par le DMA en lui envoyant un signal physique Ainsi l'UC est active d'une manière permanente (Figure II 3).
- ⇒ La notion de recouvrement n'a été rendue possible que par l'introduction du mécanisme d'interruption,
- ⇒ 1969 : Unix Operating System par Ken Thompson and Dennis Ritchie à Bell Laboratoires.

- ⊗ Problème : le cas d'un système multi utilisateur ?

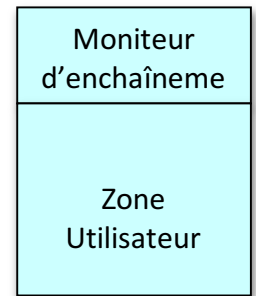


Figure II-2 : Moniteur d'Enchaînement

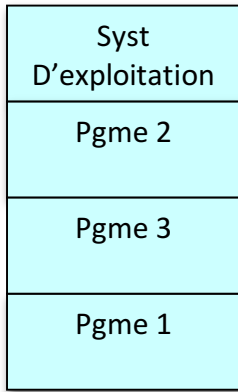


Figure II-3-a : la multiprogrammation

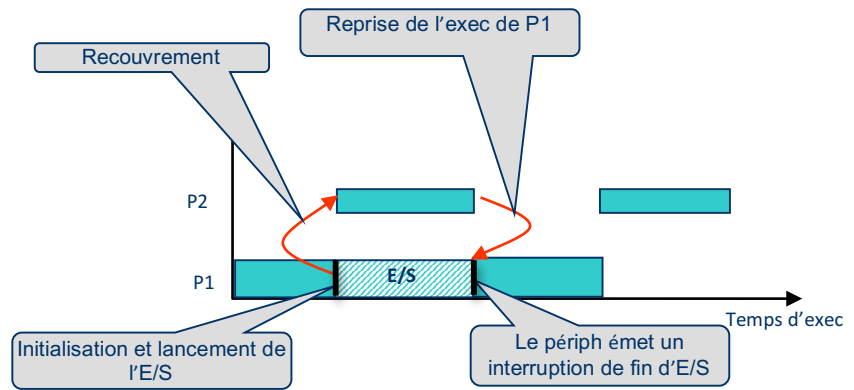
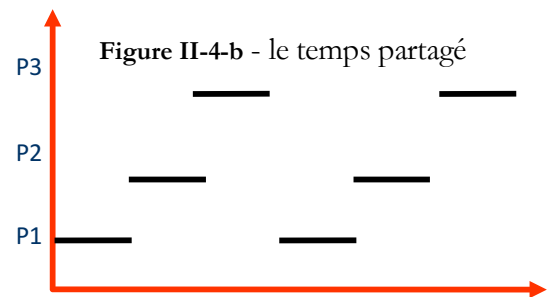
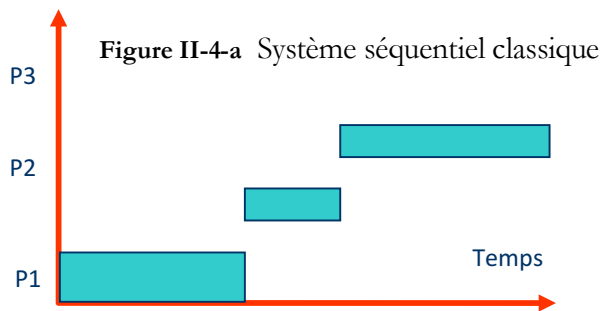


Figure II-3-b Recouvrement d'E/S

**3.2. La Multiprogrammation et le multiutilisateur : le système à temps partagé**

Plusieurs utilisateurs travaillent simultanément sur une machine, chacun avec ses propres programmes et



son propre terminal. De plus, le processeur ne peut exécuter à un moment donné qu'un seul programme à la fois. Pour satisfaire tous les programmes, l'UC est attribuée successivement à chacun d'eux pendant une fraction de temps bien déterminée. Lorsque cette période est bien étudiée, l'utilisateur aura l'impression qu'il est seul devant la machine (copies virtuelles de l'UC).

⇒ Les SE d'exploitation Linux et Windows sur PC monoprocesseur sont des systèmes à temps partagé.

**4. Les PC, les systèmes multiprocesseurs et réseaux : 1980 - €**

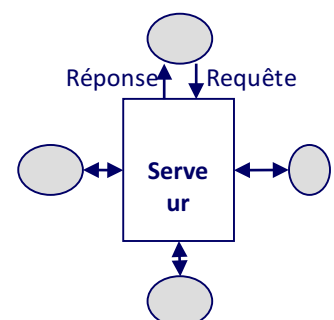
- L'apparition des micro-ordinateurs et les systèmes interactifs : les PC et MS DOS.
- L'apparition des machines multiprocesseurs.

Les systèmes réseaux regroupent plusieurs ordinateurs éventuellement géographiquement éloignés. Dans de tels systèmes, les utilisateurs sont conscients de l'existence de plusieurs ordinateurs et peuvent accéder à des machines distantes pour copier des fichiers (par exemple) d'une machine a une autre. Chaque machine a sa propre SE et ses propres utilisateurs.

⇒ 1991 : windows 3.1 - 1992 : Linux par Linus Torvald; 1993 Windows NT par Microsoft

**4.1.1. Le Modèle Client Serveur**

C'est un système réseau où l'on a des machines clientes contactant un serveur, une machine généralement très puissante en terme de capacités d'entrée-sortie, qui leur fournit des services. Ces services sont des programmes fournissant des données telles que l'heure, des fichiers, une connexion, des données, ... (sreur de BDD, sezrveur de messagerie, ...)



- ⇒ Les clients ne peuvent voir que le serveur
- ⇒ Centralisation des données et leurs traitement (requêtes-réponses)
- ⇒ Sécurité accrue

**Remarques**

1. Ces systèmes ne diffèrent pas d'un SE monoprocesseur. Ils ont besoin d'une interface réseau et quelques drivers de bas niveau pour les piloter et éventuellement quelques programmes pour les accès distants. Ces ajouts ne modifient pas la structure essentielle du noyau du SE monoprocesseur.
2. Le serveur constitue le maillon faible du système.

4.1.2. Les Systemes distribués

Dans de tels systèmes, qui sont aussi des systèmes réseaux, l'utilisateur n'est pas conscient où se trouvent ses fichiers ni où s'exécutent ses programmes. Le SE distribué se charge automatiquement et efficacement de lui ramener toute donnée dont il a besoin et exécuter tout programme qu'il demande. L'utilisateur a l'air d'être assis devant une machine locale monoprocesseur, alors que ses programmes s'exécutent réellement par d'autres UC dans le réseau et ses données proviennent d'autres endroits. Mais la situation est encore plus complexe qu'un système monoprocesseur.

Les systèmes distribués se caractérisent par :

- Partage de ressources
- Accélération du calcul
- Fiabilité (absence de maillon faible)
- Communication (échange de données)

4.1.3. Les systemes temps réel

Les Systemes Temps réels sont des systèmes où le temps joue un facteur très important.

**Exemple** : contrôle de procédés industriels.

**Caractéristiques :**

- Sont des systèmes généralement dédiés.
- Les ordinateurs sont généralement assez puissants.
- Temps de réponse immédiat et assez bref.
- Système d'interruption assez performant.
- Il y a des évènements critiques.

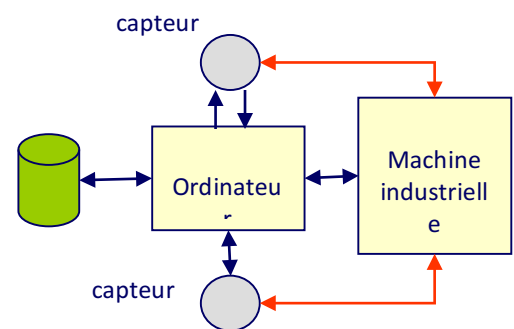
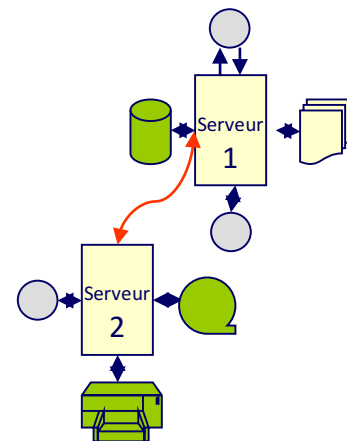
4.1.4. Les Systemes Embarqués

C'est un autre type de systèmes d'exploitation contrôlant des architectures à microprocesseur réduites en taille et en fonctions. On peut les trouver essentiellement dans les PDA, les portables téléphoniques, les lecteurs vidéos, ...

Un exemple typique d'un SE embarqué : PalmOS , Windows CE, Linux Embarqué.

**5. Terminologie**

- Système Mono tâche (mono programmé) : présence d'un seul programme utilisateur dans la MC.
- Système Multitâches (multiprogrammé) : présence de plusieurs programme utilisateur dans la MC
- Système Multitraitement : la machine est dotée de plusieurs UC
- Système Multiutilisateurs : système permettant à plusieurs utilisateurs de travailler simultanément sur une machine.

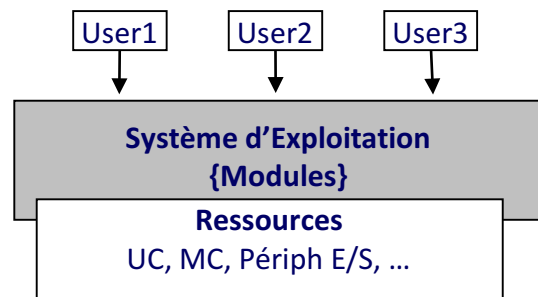


### III. Machine Virtuelle

De cette historique on peut conclure que les objectifs principaux d'un SE est de s'interfacer entre l'utilisateur et la machine pour construire au dessus de cette machine physique une machine virtuelle constituée de plusieurs couches :

- Plus pratique (Facile et commode) à l'emploi par les utilisateurs
- Plus efficace (Optimalité) pour la gestion des ressources du SIQ.

⇒ Le SE à la charge de rendre la machine facilement et efficacement exploitable



### IV. Conclusion

- Un système d'exploitation est un ensemble de programmes agissant comme interface entre l'utilisateur et le matériel. Son but principal est de fournir à l'utilisateur une machine virtuelle plus pratique à utiliser en optimisant la gestion et le contrôle des ressources.
- Les SE et le matériel évoluent d'une manière interactive
- Aujourd'hui, on a tendance beaucoup plus à des systèmes réseaux plus performants dont la sécurité est le problème primordial

### V. Références :

- Principes des systèmes d'exploitation, Abraham Silberchatz
- Les Systèmes d'Exploitation modernes, Andrew S. Tanenbaum. 2nd edition
- Systèmes d'exploitation, Joffroy Beauquier
- Systèmes d'exploitation, Mécanismes de bases, A. Belkhir