

Chapitre 01 : Introduction aux Systèmes d'Exploitation₁(SE)

1. Définitions de SE :

1

Il existe plusieurs définitions possibles des systèmes d'exploitation.

Définition 01 : Un SE est un ensemble de programmes qui coopèrent à la gestion des ressources de la machine (ordinateur).

Définition 02 : Un SE est la partie logicielle d'ordinateur qui contrôle le matériel - d'ordinateur - et fournit une base pour les programmes utilisateurs.

2. Fonctions d'un SE

L'objectif principal d'un SE est de rendre le système informatique plus pratique à utiliser en présentant à l'utilisateur une machine virtuelle à la place de la machine réelle et en gérant les ressources d'une manière transparente :

a) Gestionnaire de ressources : Le SE permet l'ordonnancement et le contrôle de l'allocation des processeurs, des mémoires et des périphériques d'E/S entre les différents programmes qui y font appel.

Exemple : Supposons que deux processus utilisateurs P1 et P2 veulent lancer simultanément l'impression de leurs résultats. Donc, ces deux processus vont utiliser la même ressource physique à savoir l'imprimante. Si le contrôle et la gestion de l'imprimante ne sont pas assurés par le SE, nous risquons d'avoir les résultats du processus P1 mélangé avec ceux de P2.

b) Présenter une machine virtuelle à l'utilisateur : Son rôle est de masquer des éléments fastidieux liés aux matériels comme les interruptions, les horloges, la gestion de la mémoire,...

3. Organisation en couches d'un SE

Le système d'exploitation est structuré en couches. Chaque couche utilise les fonctions des couches inférieures. La figure suivante montre cette structuration en couche d'un SE.

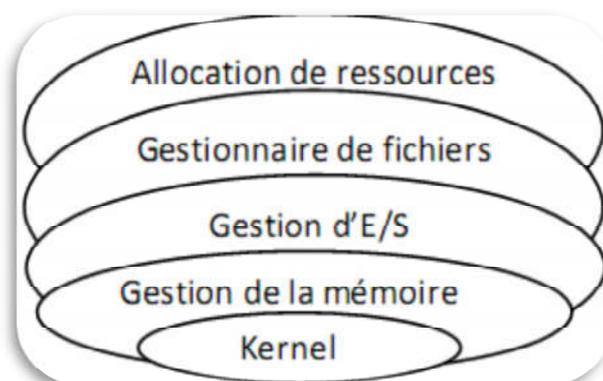


Figure 1 - structuration en couche d'un SE

- Kernel (noyau): l'interface entre le matériel et le logiciel. Il gère les interruptions et les processus (la communication et la synchronisation).
- Gestionnaire de la mémoire : Il se charge du partage de la mémoire entre les processus en attente d'exécution.
- Gestion d'E/S : Il se charge de gérer tous les périphériques (clavier, écran, disques, ...).
- Gestionnaire de fichiers : Il se charge de la gestion de l'espace du disque, de la manipulation des fichiers.
- Allocation de ressources : Il se charge d'assurer une bonne utilisation des ressources ; de fournir des statistiques sur l'exploitation des ressources. En plus, il permet à chaque processus existant dans le système d'obtenir les ressources nécessaires dans des limites de temps raisonnables et d'éviter les situations de blocage.

4. Position du système d'exploitation dans un système informatique

Un système informatique est un ensemble de matériels et logiciels destinés à réaliser des tâches qui mettent en jeu le traitement automatique de l'information. La figure 2 présente l'architecture d'un système informatique en modèle en couches. Il est généralement composé de deux couches :

- Couche du niveau 0 : qui représente la machine physique c'est-à-dire tous ce qui est matériel
- Couche du niveau 1 : qui représente tous ce qui est software. Les logiciels se répartissent également en deux grandes catégories :
 - Le système d'exploitation qui permette le fonctionnement de la machine.
 - Les programmes d'application qui résolvent les problèmes des utilisateurs.



Figure 2 - Structure en couches d'un système informatique.

5. Évolutions des systèmes d'exploitation

🕒 **La porte ouverte [1945-1950]:** Les premiers systèmes informatiques ne disposaient pas de systèmes d'exploitation (i.e. Les programmes étaient écrits directement en langage machine). Initialement, les périphériques des ordinateurs se résument en un lecteur de cartes, un perforateur de cartes, et une imprimante. Chaque programmeur disposait de son programme sous forme de paquet de cartes perforés. Dans ce cas, l'utilisateur assure le rôle de l'opérateur. En effet, il doit réaliser les opérations suivantes :

- (1) - Placer les cartes du programme source dans le lecteur ou l'unité d'entrée
- (2) - Initialiser un programme de lecture de carte
- (3) - Lancer la compilation du programme source
- (4) - Placer les cartes données (s'il y en a) dans le lecteur de cartes
- (5) - Initialiser l'exécution du programme compilé
- (6) - Extraire les résultats de l'imprimante.

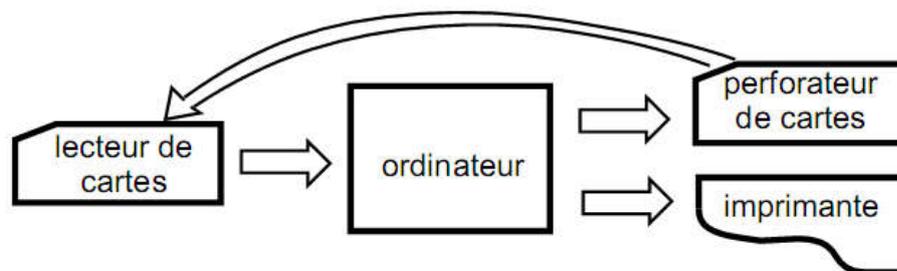


Figure 3 - Fonctionnement d'ordinateur dans première génération (Portes ouvertes)

Critiques:

- Ordinateur = grande machine, peu fiable.
- Utilisation d'ordinateur par réservation
- Pour faire exécuter un programme, l'utilisateur doit intervenir dans tous les tâches
- un seul programme à la fois est exécuté par l'ordinateur.

🕒 **Le moniteur d'enchaînement [1950-1955]:** Une amélioration du système précédent consiste à faire prendre en charge par la machine, sans intervention de l'opérateur, les opérations : lire, compiler, charger et exécuter les programmes. Le rôle de l'opérateur est alors réduit à charger les cartes et d'en extraire les résultats.

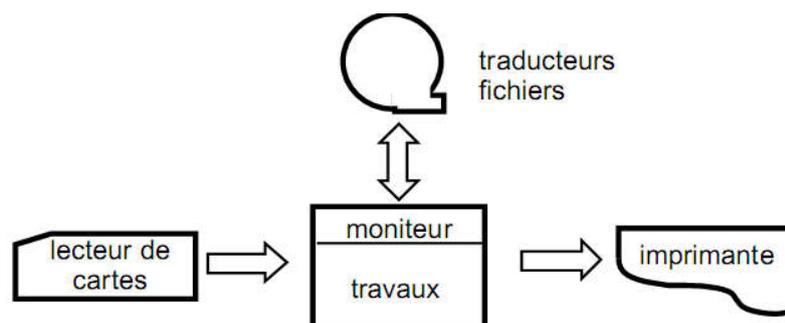


Figure 4 - Le moniteur d'enchaînement

🕒 **Traitement par lots (La Monoprogrammation) [1950-1965]:** Dans ces systèmes, il y'a introduction de la notion de lot (série de travaux). L'opérateur regroupe une suite de travaux en un lot qui sera soumis à la machine. Quand le contrôle indique l'exécution d'un programme, le moniteur d'enchaînement charge le programme et lui transfère le contrôle. Une fois terminé, le programme redonne le contrôle au moniteur d'enchaînement qui continue avec la prochaine carte de contrôle. Ces actions sont répétées jusqu'à l'exploration de toutes les cartes du job courant. Ensuite, le moniteur passe au job suivant.

Critique des systèmes de traitement par lot :

- Manque d'interaction entre l'utilisateur et le travail pendant l'exécution
- Lenteur des opérations d'E/S

⌚ **La multiprogrammation [1965-1970]:** L'objectif de la multiprogrammation est de mieux utiliser le processeur pendant les entrées/sorties (qui sont gérées par les unités d'échange). En

effet, un travail en attente peut utiliser le processeur libéré par un travail en attente d'entrée/sortie. Ceci nécessite la présence simultanée de plusieurs programmes en mémoire. Par conséquent, un mécanisme de gestion de la mémoire est indispensable.

⌚ **Le temps partagé[1970-1980]:** C'est une généralisation de la notion de multiprogrammation dans un système interactif. Ce mode consiste à offrir à chaque utilisateur l'équivalent d'une machine virtuelle tout en le faisant bénéficier de services communs. En effet, l'unité centrale est multiplexée entre les terminaux ; chacun d'eux est servi pendant un « quantum » de temps d'où son nom : temps partagé (time sharing).

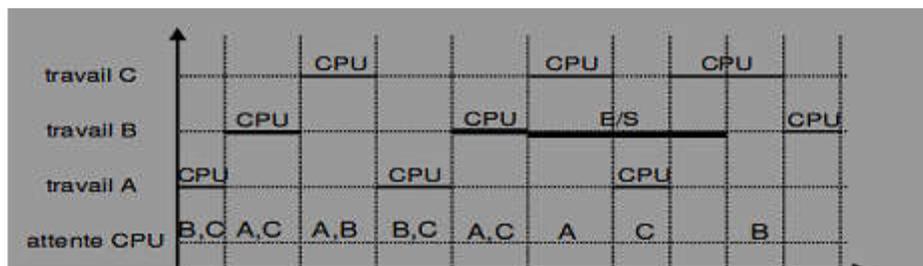


Figure 5 - schéma générale d'exécution des travaux dans un système en temps partagé.

Exemple :

6. Notions générales

❖ **Le système d'exploitation DOS [1980-]:** Le DOS (Disk Operating System) est un système d'exploitation mono-utilisateur qui apparut en 1980. Il est composé de trois principaux éléments : le BIOS qui gère les entrées/sorties de base, le noyau DOS qui contient les programmes d'accès aux fichiers et l'affectation de la mémoire et le processeur d'instruction qui constitue l'interpréteur de commandes.

❖ **Unix/Linux [1970-1991]:** Unix est un système de gestion de fichiers arborescent avec un shell (Interpréteur de commandes) conçu comme processus utilisateur. LINUX est système d'exploitation, basé sur Unix, proposé par Linus TORVALDS en 1991. Le code source de Linux est écrit pour l'essentiel en C, est largement accessible.

❖ **Le système à 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.

❖ **Le système distribué :** C'est un ensemble de processeurs indépendants, interconnectés, mais qui peuvent tous coopérer dans le cadre d'une même application. Le système se présente aux utilisateurs comme un système d'exploitation centralisé, mais qui s'exécute sur des processeurs indépendants.

7. Les différentes classes de SE

❖ Selon les services rendus

✓ Mono/Multi-tâches : capacité du système à pouvoir exécuter plusieurs processus simultanément. Par exemple, effectuer une compilation et consulter le fichier source du programme. Exemple : Mono-tâche (DOS) ; Multi-tâches (Unix, WindowsXX,...).

✓ Mono/Multi-utilisateurs : capacité à pouvoir gérer un panel d'utilisateurs utilisant simultanément les mêmes ressources matérielles. Exemple : Mono-utilisateur (WindowsXX) ; Multi-utilisateurs (Unix).

❖ Selon leur architecture :

✓ Systèmes centralisés : L'ensemble du système est entièrement présent sur la machine considérée. Par conséquent, le système ne gère que les ressources de la machine sur laquelle il est présent. Ex : Unix, Windows.

✓ Systèmes répartis : Les différentes abstractions du système sont réparties sur un ensemble de machines. Quand l'utilisateur lance un programme, il ne sait pas le nom de la machine qui l'exécutera.

❖ Selon l'architecture matérielle qui les supporte :

✓ Monoprocasseur (Ressource processeur unique): Il a fallu développer un mécanisme de gestion des processus pour offrir un (pseudo) parallélisme à l'utilisateur : c'est la multiprogrammation.

✓ Multiprocesseurs : Ces systèmes sont composés de plusieurs processeurs reliés au bus de l'ordinateur. Ils se caractérisent par leur capacité de traitement et leur fiabilité.