

Matière : systèmes d'exploitation 1 (Operating System)
 Chapitre I : Introduction aux systèmes d'exploitation

1. Introduction

Les trois éléments essentiels d'un ordinateur sont le processeur, la mémoire et le dispositif de gestion des entrées-sorties. Ils communiquent entre eux par l'intermédiaire du bus.

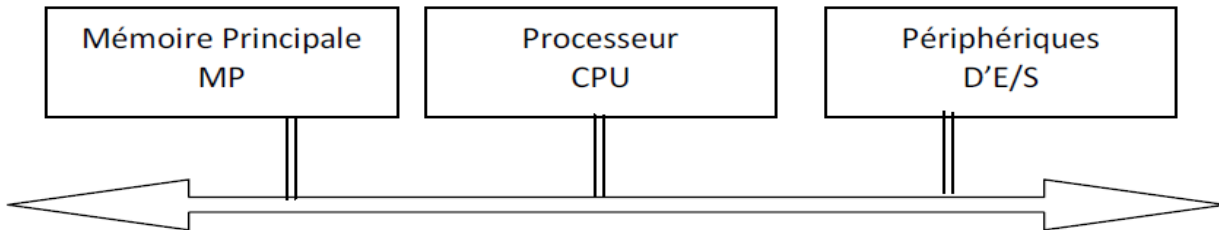


Figure 1 : Architecture générale d'un ordinateur-Von Neumann

Le problème fondamental est de réussir à faire fonctionner cet ensemble, à assurer les échanges d'information entre les différents éléments qui constituent la machine et surtout, du point de vue de l'utilisateur, à dialoguer avec lui. C'est le rôle du système d'exploitation.

2. Définition d'un S.E

Il existe plusieurs définitions possibles des systèmes d'exploitation.

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

Définition 02 : Un S.E peut être vu comme un programme agissant en tant qu'intermédiaire entre l'utilisateur et le matériel de l'ordinateur.

Définition 03 : le SE est un logiciel dont l'objectif est de libérer l'utilisateur d'un ordinateur de la complexité de la programmation du matériel. Il propose une gestion flexible et optimisée de ses différents composants (processeur, affichage, stockage, etc.) .

Définition04 : un SE est la partie principale dans un système informatique. Il permet d'exploiter les caractéristiques de la machine physique sans se soucier des détails de celle-ci.

Définition 05 : un SE ou OS est un ensemble de programmes qui assurent la gestion et le contrôle des ressources d'un ordinateur. [1]

Définition 06 : Le système d'exploitation est un ensemble de programmes qui permettent le fonctionnement de l'ordinateur. Son objectif principal est de présenter une machine virtuelle à l'utilisateur. Ainsi de gérer les ressources matérielles (mémoire, processeur,...).

Le système d'exploitation est en quelque sorte le coordinateur des interactions entre les matériels, les logiciels et les utilisateurs.

Exemple: Quand on appuie sur une touche du clavier c'est le système d'exploitation qui reçoit cet événement et le renvoie au bon processus.

Nous pouvons dire que :

$$S.E = \text{Programme d'allocation} + \text{Programme de contrôle}$$

Définition d'un SI : un système informatique est composé d'une : matériel (la machine physique ou HARDWARE) , SE, des applications, des utilisateurs.

Exemples :

Unix : Créé en 1969, rapidement multi-utilisateur, écrit en langage C.

Linux : Clone gratuit d'UNIX pour les PC, open source.

Mac OS : Premier à proposer le concept des fenêtres, aujourd'hui possède le noyau Linux.

MS-DOS (Microsoft disque operating system) : SE des premiers PC, Mono-utilisateur, mono-tâche, interface ligne de commande.

MS-Windows. ; SUNOS ; BDS, Solaris, Android et ios (tablettes ou téléphones portable), QNX (systèmes robotiques)

3. Historique

1ère génération (1945-1955)

- Pas de système d'exploitation
- Langage machine

2ème génération (1955-1965)

- Systèmes d'exploitation : FMS, IBYS
- Langage assembleur et fortran

3ème génération (1965-1980)

- Systèmes d'exploitation : MULTICS, UNIX, Linux, ...
- Nouveau concept : multiprogrammation

4ème génération (1980-aujourd'hui)

- Systèmes d'exploitation : Macintosh, Windows, ...
- Nouveau concept : IHM graphique.

1. Ressources gérées par un SE : Quelles sont ? Définition: Nous appelons ressource toute entité nécessaire à la progression d'un processus.

Un ordinateur est constitué de deux éléments principaux :

1. **Le matériel (Hardware)** : qui fournit les ressources physiques. On retrouve : le(s) processeur(s), la(les) mémoire(s), et le(s) périphérique(s) d'Entrée/Sortie (E/S).
2. **Le logiciel (Software)** : qui exploite matériel pour remplir les fonctions attendues d'un ordinateur. On peut trouver : *les programmes d'application des utilisateurs*, comme les traitements de texte, logiciels de comptabilité, les applications de gestion bancaire, etc. *Les données des utilisateurs*, comme les bases de données.

Opérations sur les ressources

Chaque type de ressource est associé à un ensemble d'opérations que l'on peut lui appliquer. Cependant, Ils possèdent tous les opérations communes suivantes :

- Demander
- Allouer
- Utiliser
- Libérer

Dans certains cas, on autorise une opération supplémentaire sur certaines ressources : **la réquisition**. Ce cas se produit quand nous avons une ressource allouée à un processus et si une demande survient de la part d'un processus **plus prioritaire**

2. Fonctions d'un S.E

L'objectif (rôle) principal d'un S.E 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** (Le SE présente au programmeur une interface d'accès aux ressources de l'ordinateur (sous forme d'appels système). Ainsi le programmeur peut faire abstraction des détails de fonctionnement des ressources. Cette interface est fondée sur des objets abstraits dont les plus importants sont les fichiers et les processus. Par exemple, le programmeur voit un disque comme une collection de fichiers qui peuvent être lus, écrits et fermés). Le S.E 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 S.E, nous risquons d'avoir les résultats du P1 mélangé avec ceux de P2.

Remarque : *Machine virtuelle*[2]

- Masquer les éléments fastidieux du matériel pour fournir une interface simple à utiliser.
- Permettre à plusieurs programmes de fonctionner en même temps, chaque programme ayant sa propre machine virtuelle. Le système d'exploitation permet de maintenir cette illusion.

3. Tâches (rôles) d'un système d'exploitation

- *Gestion de processus* : le système d'exploitation est chargé de gérer l'allocation du processeur entre les différents programmes grâce à un algorithme d'ordonnancement,
- *Gestion de la mémoire* (gérer l'utilisation de la RAM) : le système d'exploitation est chargé de gérer l'espace mémoire alloué à chaque application. En cas d'insuffisance de mémoire physique, le système d'exploitation peut créer une zone mémoire sur le disque dur, appelée «mémoire virtuelle». La mémoire virtuelle permet de faire fonctionner des applications nécessitant plus de mémoire qu'il n'y a de mémoire vive disponible sur le système.
- *Gestion des Entrées/Sorties* (gérer les opérations d'E/S) : le système d'exploitation permet de contrôler l'accès des programmes aux ressources matérielles par l'intermédiaire des pilotes (*appelés également gestionnaires de périphériques ou gestionnaires d'entrée/sortie*).
- *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.
- *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 utilisateurs possédant les droits adéquats.
- *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.

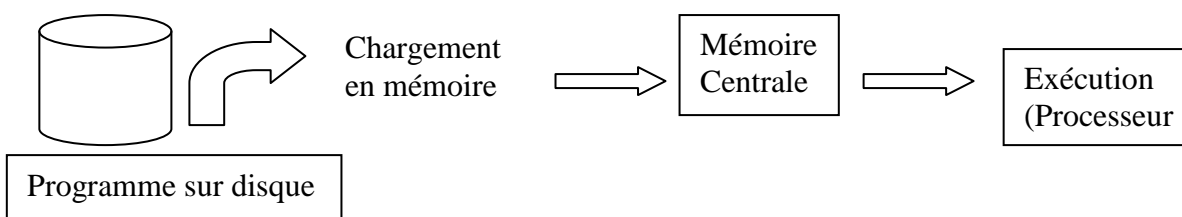
Les programmes utilisateurs peuvent accéder à ces différentes fonctionnalités à l'aide **des appels système**.

Pour créer un système aussi grand et complexe qu'un SE, il est nécessaire de le découper en pièces plus petites.

a. Contrôle d'exécution des programmes :

Un programme écrit en un certain langage évolué nécessite d'abord d'être traduit en langage machine (code binaire), avant d'être exécuté. Une fois traduit, le programme sera chargé ou copié dans un espace contigu de la mémoire centrale. L'exécution du programme se fera alors en décodant chaque instruction, puis en l'exécutant effectivement en la soumettant à l'unité de calcul du processeur central.

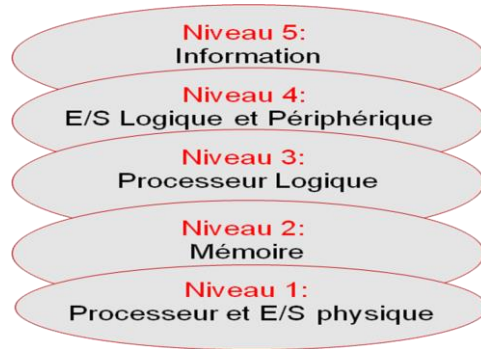
Ces différentes étapes sont prises en charge par des programmes spécifiques du système d'exploitation, sans faire appel à la connaissance de l'utilisateur.



On dit que le SE alloue la ressource processeur à un des programmes usager. Tous ces choix doivent se faire par des programmes du SE., en donnant l'impression à l'utilisateur qu'il utilise seul la machine, et en lui minimisant au mieux le temps de réponse des programmes lancés.

4. Structure Hiérarchique d'un S.E

La structure hiérarchique d'un système d'exploitation, selon E.W. Dijkstra, est montrée par la figure suivante :



Niveau1 : Ce niveau est en relation directe avec la machine (i.e. Matériel). Il s'occupe de l'allocation et libération du processeur aux programmes usagers (à qui, quand, combien de temps), ainsi que l'exécution des entrées/sorties physiques.

Niveau2 : Ce niveau s'occupe de l'allocation/libération de la mémoire centrale aux programmes usagers (à qui, quand, pour combien, à quelle adresse).

Niveau3 : L'allocation du processeur aux programmes usagers, nécessite de les organiser d'abord, puis de gérer les différents états par lesquels il passe. Cette gestion est dite *gestion du processeur logique*. En résumé, ce niveau s'occupe de la création, destruction, activation, suspension, synchronisation et communication entre *processus*.

Niveau4 : Il se charge à la gestion des *E/S logiques*, maintient des états des périphériques, allocation et libération des périphériques.

Niveau5 : Ce dernier niveau gère l'information. Ainsi, il s'occupe de la création, destruction, ouverture, fermeture, lecture et écriture des fichiers.

5. Classes de systèmes d'exploitation

Les systèmes d'exploitation peuvent être classés en catégorie, suivant les critères considérés.

① Selon les services rendus

- ❖ **{mono|multi} tâches :** Multitâches (en anglais *multithreaded*): capacité du système à pouvoir exécuter plusieurs processus simultanément; par exemple effectuer une compilation et consulter le fichier source du programme correspondant. C'est le cas d'UNIX, d'OS/2 d'IBM et de Windows 95.

Les applications sont composées en séquence d'instructions que l'on appelle «**processus légers**» (en anglais «*threads*»). Ces threads seront tour à tour actifs, en attente, suspendus ou détruits, suivant la priorité qui leur est associée ou bien exécutés séquentiellement.

- ❖ **{mono|multi} utilisateurs :** Multi-utilisateurs : capacité à pouvoir gérer un panel d'utilisateurs utilisant simultanément les mêmes ressources matérielles. C'est le cas d'UNIX et MVS.

② Selon l'architecture matérielle qui les supporte

- ❖ **Architecture mono-processeur (*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.
- ❖ **Architectures multiprocesseurs (*parallélisme*)** : Ces systèmes sont composés de plusieurs processeurs reliés au bus de l'ordinateur. Ils se caractérisent par leur capacité de traitement et leur fiabilité i.e. a panne d'un processeur n'arrêtera pas le système. 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.
- ❖ **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 :
 - PalmOS
 - Windows CE / Windows Mobile / Windows Smartphone

6. Evolution des systèmes d'exploitation

6.1 La porte ouverte : [3]

Les premiers systèmes informatiques *ne disposaient pas de systèmes d'exploitation*. L'ordinateur, dépourvu de tout logiciel, était utilisé à tour de rôle par les utilisateurs. Chaque programmeur disposait de son programme sous forme de paquet de cartes perforés, de rubans de papiers ou de bandes magnétiques. Dans ce cas, l'utilisateur assure le rôle de l'opérateur ; 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 cartes.
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.

Inconvénients :

- Temps perdu dans l'attente pour lancer l'exécution d'un programme.
- Vitesse d'exécution de la machine limitée par la rapidité de l'opérateur qui appuie sur les boutons et alimente les périphériques.
- seul l'enchaînement des instructions d'un programme est automatique.

6.2 Le moniteur d'enchaînement :

Une amélioration du système précédent consiste à faire prendre en charge par la machine (le moniteur d'enchaînement), 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 à une extrémité de la machine et d'en extraire les résultats sur papier de l'autre.

Inconvénients :

- Ce système est que celui-ci doit pouvoir réagir en fonction des cartes de contrôle. Il doit être capable d'interpréter un langage de contrôle de job (programme).
- D'autre part, la machine est occupée pendant l'entrée des données ou la sortie des résultats (le système est principalement limité par les performances d'E/S).

6.3 Traitement par lots (Batch):

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.

Exemple: cartes de contrôle: %FTN: exécuter le compilateur FORTRAN; %ASM exécuter l'assembleur; %RUN: exécuter le programme courant; %JOB: première carte d'un job; %END: dernière carte.

6.4 La multiprogrammation (parallélisme):

Pour pallier aux inconvénients du moniteur d'enchaînement, de maintenir en mémoire plusieurs travaux ou jobs ou processus prêts à exécuter, et partager efficacement les ressources de la machine entre ces jobs. Une technique fondamentale est apparue : la multiprogrammation.

La multiprogrammation vise à *augmenter l'utilisation du processeur* en ayant toujours une occupation pour celui-ci. Ceci nécessite la présence simultanée de plusieurs programmes en mémoire.

La multiprogrammation (ou multi-tasking en anglais) apporte une solution au problème d'attente des *processus*, et la perte d'utilisation du processeur qui en découle. Dans ce contexte, nous allons considérer que le processus est *l'unité d'exécution principale* correspondant à une tâche qui doit être effectuée par le processeur.

6.4.1 Remarque

S'il est courant qu'un programme correspond à un seul processus, il n'y a pas d'équivalence générale entre processus et programme : un programme peut être constitué de plusieurs processus collaboratifs, c'est-à-dire de plusieurs processus échangeant entre eux de l'information afin de réaliser ensemble une tâche donnée.

| monoprogrammation | multiprogrammation |
|--|---|
| <ul style="list-style-type: none">• SE simple• Mauvaise utilisation des ressources (mémoire, processeur, E/S)• Temps de réponse imposé par les travaux longs | <ul style="list-style-type: none">• SE plus complexe.• Possibilité de mieux équilibrer la charge des ressources.• possibilité d'améliorer le temps de réponse pour les travaux courts |

6.4.2 Le temps partagé (Time Sharing)

Si un programme de courte durée est soumis après des programmes de longue durée, il sera lésé et attendra jusqu'à ce que le processeur soit libéré par les programmes qui le précèdent.

C'est la généralisation de la notion de multiprogrammation dans un système interactif: conversationnel. Ce mode consiste à offrir à chaque utilisateur une machine "virtuelle" qui lui est propre. L'utilisateur se croit l'occupant de

toutes les ressources de la machine. Ce mode privilégie l'interaction Homme-Machine. L'UC est multiplexée entre les terminaux; chacun d'eux est servi pendant un "**quantum**" de temps d'où son nom: temps partagé (time sharing).cette conception a nécessité l'introduction d'un dispositif matériel qui comptabilise le temps nommé une horloge temps réel.

Ainsi, une *interaction directe* avec la machine est offerte aux usagers travers des *terminaux de conversation*. L'utilisateur peut alors contrôler le programme (dit aussi travail ou job) qu'il a soumis directement à partir du terminal (corriger les erreurs ; recompiler, re-soumettre le job, ...).

En fait, la plupart des systèmes d'aujourd'hui sont en temps partagé.

6.5 Systèmes à temps réel :

Est appelé temps réel le comportement d'un système informatique dont le fonctionnement est assujéti à l'évolution dynamique d'un procédé industriel connecté à lui. On dit que le processus est contrôlé, piloté ou supervisé par le système qui réagit aux changements d'état du processus.

Ce type de systèmes est un SE spécialisé, dédié à des applications spécifiques, en particulier des systèmes de contrôle, pourvus de capteurs. Ceux-ci captent de l'information qu'ils fournissent à l'ordinateur, puis celui-ci analyse ces informations, réalise les contrôles et donne les résultats pour d'éventuelles interventions. Voici quelques exemples de systèmes d'exploitation temps réel : · OS-9 ; RTLinux (RealTime Linux) ; · QNX ; · VxWorks.

Exemples :

- Chaîne industrielle robotisée.
- Contrôle d'un réacteur nucléaire : dans le cas de chauffage du réacteur, des procédures doivent être prises avant que le réacteur n'atteigne une température élevée.
- Systèmes d'imagerie médicale... etc.

6.6 Systèmes distribués

Un système distribué est un ensemble d'entités autonomes de calcul (ordinateurs, PDA, processeurs, processus, processus léger etc.) indépendants, interconnectés et qui peuvent communiquer et coopérer dans le cadre d'une même application.

- Exemples: – réseau physique de machines – Un logiciel avec plusieurs processus sur une même machine.

Exemples: • WWW • Contrôle du trafic aérien • Système de courtage

- Banques • Super calcul distribué • Système de fichier distribué
- DNS • Systèmes Pair-à-pair (P2P)

Bibliographie

[1] : principe systèmes d'exploitation. N.SALMI. pages bleues 2007. (univ khemis : code :004-363)

[2] : Système d'exploitation. Cours 1. Jean-Sébastien Coron, Université du Luxembourg.

[3] : Dr BELKHIR Abdelkader. « Système d'exploitation Mécanisme de base». 2005. [004-255]

[4] : Systèmes d'exploitation. Andrew Tanenbaum, 2^o édition 2003. (univ khemis : code :004-082).

Enseignant : Omar boukadoum / Email : boukadoum2020@gmail.com
Page Facebook : <https://www.facebook.com/boukadoumomar/>