

## CHAPITRE 1

# Introduction aux Systèmes d'Exploitation

### I.1. Introduction

Les concepteurs visaient par la création du premier ordinateur de faciliter la vie de l'être humain en automatisant tout type de tâche. Vers les années 40, un seul groupe de personnes concevait, construisait, programmait et utilisait la machine à cause de sa complexité (d'un autre terme, l'utilisation de l'informatique était limitée dans ce groupe), aux cours des années suivantes cette complexité est devenue réduite petit à petit jusqu'aux années 80 où l'ordinateur est évolué pour devenir à la portée de tout le monde tout en utilisant un élément intermédiaire qui doit s'acquitter de toutes les tâches.

Sans logiciels, un ordinateur n'est qu'un morceau de matériel inutile. Grâce à ces logiciels, il peut mémoriser, traiter et restituer des informations et servir dans de nombreuses autres activités qui justifient son utilité.

Les logiciels se répartissent en deux grandes catégories:

- Les programmes système qui permettent le fonctionnement de l'ordinateur.
- Les programmes d'application qui résolvent les problèmes des utilisateurs.

Il y a quelques années, il est apparu clairement qu'il fallait trouver un moyen de libérer les programmeurs de la complexité du matériel. Ce moyen, qui a évolué petit à petit, consiste à enrober le matériel par une couche de logiciel qui gère l'ensemble de système. Il va falloir en fait présenter au programmeur une interface ou machine virtuelle plus facile à comprendre et à programmer. Cette couche de logiciel est le système d'exploitation.

Le système d'exploitation 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. La situation est présentée par la figure 1.1.

Système bancaire	Réservation d'avion	Contrôle de jeux processus industriel	Programmes d'application
Compilation	Edition	Interpréteur de commande	Programmes système
Système d'exploitation			
Langage machine			Matériel
Microprogrammes			
Dispositif physique			

Figure 1.1 : Organisation d'un système informatique

Le matériel se trouve au niveau le plus bas et se compose lui-même de deux couches ou plus. La couche la plus basse contient les circuits physiques constitués de circuits intégrés, de fils électriques, d'alimentation, de tube cathodique et de périphériques physiques. De cette nature, leurs constructions et leurs fonctionnements concernent l'ingénieur en électricité. Vient ensuite un logiciel de base qui contrôle ces différents périphériques pour fournir une interface plus simple à la couche suivante. Ce logiciel, appelé un microprogramme, se situe généralement dans la mémoire morte « ROM ». Il s'agit en fait d'un interpréteur qui recherche des instructions en langage machine comme ADD, MOV et JMP et les exécute l'une après l'autre.

Le Microprogramme ne fait pas réellement partie du matériel, mais il est toujours décrit dans les manuels des constructeurs d'ordinateurs. Ce qui amène de nombreuses personnes à le considérer comme faisant partie de la « Machine ». Sur certaines machines le microprogramme est implanté dans le matériel et ne constitue pas à lui seul une couche distincte.

La fonction primordiale d'un système d'exploitation est de masquer toute la complexité du matériel et de présenter au programmeur un ensemble d'instructions plus simple à utiliser. Par exemple LIRE UN BLOC DE FICHER est conceptuellement plus simple que d'avoir à se soucier de déplacer les têtes de lecture puis à se positionner ...etc.

Le reste du logiciel système se trouve au-dessous du système d'exploitation. On y trouve l'interpréteur de commande (shell), les compilateurs, les éditeurs et les programmes de ce type qui ne dépendent pas des programmes d'application. Il faut bien se rendre compte que ces programmes ne font pas partie du système d'exploitation, bien qu'ils soient fournis par les constructeurs d'ordinateurs.

Le système d'exploitation est la partie du logiciel qui fonctionne en **mode noyau ou mode superviseur**. Elle protège le matériel contre les erreurs de manipulation des utilisateurs. Les compilateurs et les éditeurs fonctionnent en **mode utilisateur**.

En fin, on trouve les programmes d'application au-dessus des programmes système. Ces programmes sont écrits pour les utilisateurs pour résoudre leurs problèmes spécifiques tels que le traitement des données commerciales, les calculs scientifiques ...etc.

## I.2. Qu'est-ce qu'un système d'exploitation ?

Il existe plusieurs définitions possibles des systèmes d'exploitation.

### Définition 1.1

Un Système d'exploitation (SE) ou (Operating System) est la partie la plus importante d'un Système Informatique (SIQ). Il est constitué d'un ensemble de programmes agissant comme interface entre l'utilisateur et le matériel. Son but principal est de rendre le système informatique plus pratique à utiliser et qui fonctionne d'une manière efficace et optimale.

### Définition 1.2

Un système d'exploitation est un ensemble de programmes et de sous programmes (fonctions) qui assurent la gestion des ressources matérielles et logicielles pour coordonner les opérations d'un ordinateur. C'est l'intermédiaire entre les programmes d'applications et le matériel : il intercepte les demandes des applications et les transmet aux diverses ressources matérielles (mémoire centrale, périphériques d'entrée sortie,...).

Il soustrait le matériel au regard du programmeur et offre une vue agréable à l'ordinateur donc il transforme l'ordinateur en une machine virtuelle facilement manipulable par un simple utilisateur.

### Définition 1.3<sup>1</sup>

Un Système d'exploitation (SE) est un ensemble de programmes qui coopèrent à la gestion des ressources de la machine (ordinateur). Il doit satisfaire les deux fonctions (principales) suivantes :

- **Partager les ressources** : Une système doit partager les ressources de l'ordinateur entre plusieurs utilisateurs simultanés.
- **Présenter une machine virtuelle à l'utilisateur** : Le système d'exploitation doit convertir un vulgaire matériel en un machine virtuelle.

## I.3. Objectives des Systèmes d'exploitation

Les logiciels peuvent être classés en deux catégories:

- les programmes d'application des utilisateurs
- les programmes système qui permettent le fonctionnement de l'ordinateur. Parmi ceux-ci, le système d'exploitation (SE).

Le SE soustrait le matériel au regard du programmeur et offre une présentation agréable des fichiers. Un SE a ainsi deux objectifs principaux :

- **présentation** : Il propose à l'utilisateur une abstraction plus simple et plus agréable que le matériel : une machine virtuelle
- **gestion** : 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.

## I.4. Système Informatique (SIQ)

Un système informatique est généralement compose de quatre parties (composants) :

La machine physique où l'ensemble de matériels (hardware), le système d'exploitation, les programmes d'applications et les utilisateurs (Figure 1.2). Le système d'exploitation (SE) est donc une partie importante de tout système informatique.

<sup>1</sup> Livre, BELKHIR Abdelkader, Système d'exploitation mécanisme de base. (Office des Publications Universitaires).

- Le matériel : toute architecture à microprocesseur regroupant une Unité Centrale (UC), une Mémoire Centrale (MC), des périphériques d'Entrée/Sortie, .etc. et constituant l'ensemble des **ressources de base** du Système Informatique.
- Le système d'exploitation dit aussi système opératoire, est la première couche qui vient au-dessus de la couche "Matériel". C'est un logiciel de base qui fait l'objet de la première installation sur l'ordinateur. Il tourne constamment pour permettre d'utiliser et communiquer avec la machine.
- Les programmes d'applications regroupent les utilitaires aidant à la mise au point des programmes (Assembleurs, compilateurs, éditeurs de liens, ...) ainsi que les SGBD, les applications bureautiques .etc. Ces applications consomment des ressources pour pouvoir résoudre le problème visé.
- L'utilisateur c'est une entité qui demande un service de la machine, que ce soit un être humain, un ordinateur, un capteur ou une architecture à microprocesseur (robot, machine).

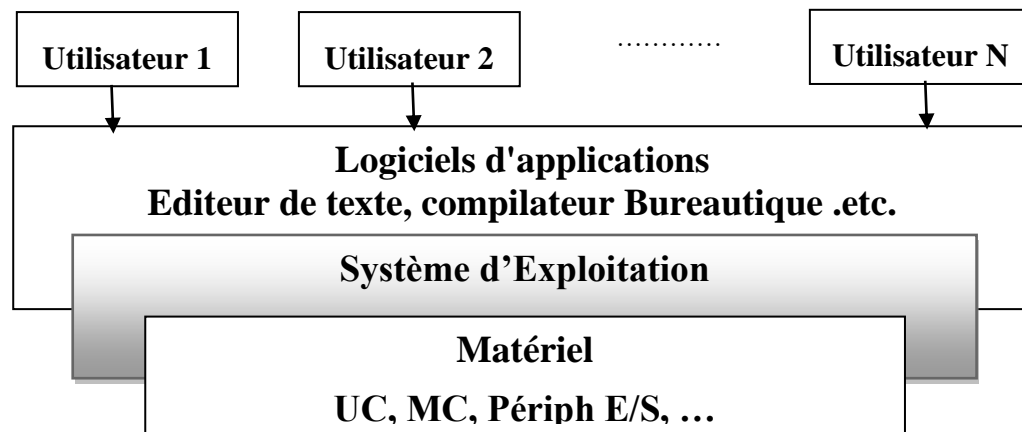


Figure 1.2 : Une vue abstraite d'un système informatique.

D'autre part, un SE peut être considéré comme **un programme d'allocation** de ressources. Un système informatique possède plusieurs ressources (matérielles et logicielles) qui peuvent être requises pour résoudre un problème donné : temps processeur, espace mémoire, espace de stockage des fichiers, périphériques d'entrée/sortie. Le SE agit en gestionnaire de ces ressources et les alloue aux programmes et aux utilisateurs quand cela est nécessaire à leur tâche. Puisqu'il peut exister plusieurs requêtes, parfois incompatibles, pour les ressources, le SE doit décider à quelle requête on doit allouer les ressources afin d'exploiter le système informatique de façon efficace et optimale.

Une vue légèrement différente d'un SE focalise sur le besoin de contrôler les divers périphériques d'Entrée/Sortie et les programmes utilisateurs. Un SE est donc **un programme de contrôle**. Ce dernier vérifie l'exécution des programmes utilisateurs pour éviter des erreurs et une utilisation incorrecte de l'ordinateur.

En conclusion, nous pouvons dire donc que :

**Un système d'exploitation = Programme d'allocation + Programme de contrôle.**

## I.5. Les rôles d'un système d'exploitation

Le système d'exploitation (Operating System, O.S.) est l'intermédiaire entre un ordinateur (ou en général un appareil muni d'un processeur) et les applications qui utilisent cet ordinateur ou cet appareil. Pour comprendre le rôle du système d'exploitation, nous allons voir le système d'exploitation selon deux points de vue: vue par l'utilisateur et vue par le système informatique.

### I.5.1. Vue par l'utilisateur (avoir un SIQ pratique)

La vue de l'utilisateur change selon l'interface utilisée. La plupart des usagers utilisent un PC (Personal Computer) composé de ressources suivantes : écran, clavier, souris et une unité. Ce système informatique est conçu pour un seul usager qui monopolise ses ressources. L'objectif c'est de maximiser le travail effectué sur ce système informatique. Dans ce cas, le système d'exploitation est conçu principalement pour faciliter l'utilisation, avec plus d'attention aux performances que de l'utilisation des ressources (comment les ressources matériel et logiciel sont partagées).

Le SE s'interface entre l'utilisateur et la machine pour fournir un environnement plus pratique à l'utilisateur et le détacher au maximum possible de toute connaissance technique du matériel en fournissant une Interface Homme Machine (IHM) plus intuitive, plus conviviale et plus facile (principe de la machine virtuelle).

### I.5.2. Vue par le système informatique (Avoir un SIQ efficace et optimale)

A partir du système informatique le système d'exploitation c'est un programme lié intimement au matériel. Dans ce contexte, nous disant que le système d'exploitation c'est **un allocateur de ressources**. Dans un système informatique il existe plusieurs ressources qui nécessitent un gouverneur, temps d'utilisation de CPU, espace mémoire, espace de stockage disque, périphérique d'E/S. Le système d'exploitation joue le rôle **d'un gestionnaire de ressources**.

Le SE doit contrôler le comportement des différentes applications des utilisateurs et celui des périphériques d'E/S. Le SE doit garder le SIQ dans un état cohérent en évitant les erreurs et les utilisations incorrectes des programmes de la machine. De plus, il doit appliquer une stratégie d'allocation des ressources (matérielles ou logicielles) rigoureuse pour en optimiser l'exploitation (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, ...).

## I.6. Les fonctions du système d'exploitation

Le système d'exploitation assure les principales fonctions :

- **Offrir une machine virtuelle simple** : il offre une machine simple à utiliser grâce à une interface qui permet d'utiliser de manière simple le matériel et qui fait abstraction des détails de sa mise en œuvre. L'utilisation du contrôleur de disquettes à titre d'exemple, suscite l'utilisation et la maîtrise de 16 commandes (lecture/écriture d'un secteur, déplacement de bras, formatage...). Chaque commande suscite la précision de plusieurs paramètres (nombre de secteurs, distance entre les secteurs, numéro de secteur...). Cette

complexité demeure transparente à l'utilisateur grâce à l'utilisation d'un système d'exploitation.

- **Gestion des processeurs:** Gérer l'allocation des processeurs entre les différents programmes grâce à un algorithme d'ordonnancement.
- **Gestion des processus (programmes en cours d'exécution) :** Gérer l'exécution des processus en leur affectant les ressources nécessaires à leur bon fonctionnement.
- **Gérer la mémoire pour exécuter des tâches :** il alloue et récupère la mémoire pour permettre l'exécution de programmes.
- **Gérer et conserver l'information :** il permet de stocker, manipuler et organiser les données dans des fichiers et répertoires.
- **Assurer l'interaction entre programmes :** il assure la communication entre les programmes pour leur permettre de collaborer et interagir entre eux.
- **Assurer la protection et gérer les erreurs :** il gère les erreurs et droits d'accès pour protéger les données des utilisateurs et les données sensibles du système.
- **Gérer et contrôler les ressources partagées :** il gère et contrôle l'utilisation de ressources partagées comme le processeur, la mémoire et les imprimantes. Autrement, il permet de répartir ces ressources entre les utilisateurs ou programmes demandeurs et d'éviter les conflits d'accès. En cas de partage d'imprimante à titre d'exemple, le système d'exploitation verrouille l'accès à l'imprimante lors de l'impression, gère les tampons d'impression, gère la file d'attente, etc.

## I.7. Historique et évolution des systèmes d'exploitation

Les systèmes d'exploitation ont été historiquement liés à l'architecture des ordinateurs sur lesquels ils étaient implantés. L'évolution de l'un des deux suit l'évolution de l'autre, ils se développent en parallèle. Le développement a été motivé par les objectifs suivants :

- Décharger le programmeur des tâches de programmations énormes et fastidieuses, et de lui permettre de se concentrer sur l'écriture de son application.
- Protéger le système des usagers et de fausses manipulations.
- Offrir une vue simple, uniforme et cohérente de la machine et de ses ressources.

Nous décrivons les générations successives des ordinateurs et observons à quoi ressemblait leur système d'exploitation.

Quatre générations de systèmes peuvent être distinguées : les premiers systèmes, les systèmes de traitement par lot, les systèmes multiprogrammés et à temps partagé.

### I.7.1. Premiers systèmes

Cette génération qui date des années 1945-1955, réfère aux premiers systèmes qui étaient des machines à tubes à vide volumineuses, très fragiles et très lentes. Ces machines destinées à la programmation en langage machine sur des cartes enfichables ou cartes perforées, étaient utilisées pour réaliser des calculs simples (calculs des tables des sinus et cosinus). Elles tournaient sans système d'exploitation et constituaient des systèmes mono-usager. L'utilisateur avait la charge de

construire programmer et maintenir son programme et devait réserver à l'avance la machine pour une plage horaire qui pouvait dépasser la plage horaire réelle dont il a besoin pour exécuter son programme. Cela conduit à une mauvaise utilisation du processeur expliquée par la perte de temps CPU qui demeure peu active.

### **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.
- Pas de différences entre : concepteurs ; constructeurs ; programmeurs ; utilisateurs ; mainteneurs.

### **I.7.2. Systèmes de traitement par lots**

La période des années 1955-1965 a connu l'utilisation des machines à transistors fiables mais qui demeurent coûteuses. Ces machines étaient destinées à la programmation en Fortran et Assembleur. Elles utilisent un chargeur et un compilateur Fortran qui constituent les premiers logiciels de base. Avec cette génération, une distinction nette entre opérateurs de la salle machine, constructeurs, programmeurs et utilisateurs a été faite.

Traitement par lots sont des systèmes réalisant le séquençage des jobs ou travaux selon l'ordre des cartes de contrôle à l'aide d'un moniteur d'enchaînement. L'objectif était de réduire les pertes de temps occasionnées par l'oisiveté du processeur entre l'exécution de deux jobs ou programmes (durant cette période, il y a eu apparition des machines à transistor avec unités de bandes magnétiques, donc évolution des ordinateurs).

L'idée directrice était de collecter un ensemble de travaux puis de les transférer sur une bande magnétique en utilisant un ordinateur auxiliaire (Ex. IBM 1401). Cette bande sera remontée par la suite sur le lecteur de bandes de l'ordinateur principal (Ex. IBM 7094) afin d'exécuter les travaux transcrits en utilisant un programme spécial (l'ancêtre des S.E. d'aujourd'hui. Ex. FMS : Fortran Monitor System, IBSYS). Les résultats seront récupérés sur une autre bande pour qu'ils soient imprimés par un ordinateur auxiliaire. Cette situation est illustrée à la Figure 1.3.

Quand le moniteur rencontre une carte de contrôle indiquant l'exécution d'un programme, il charge le programme et lui donne le contrôle. Une fois terminé, le programme redonne le contrôle au moniteur d'enchaînement. Celui-ci continue avec la prochaine carte de contrôle, ainsi de suite jusqu'à la terminaison de tous les jobs. La structure d'un travail soumis est montrée sur la Figure 1.4 :

### **Inconvénients**

- Perte de temps dû à l'occupation du processeur durant les opérations d'E/S. (En effet, le processeur restait trop inactif, car la vitesse des périphériques mécaniques était plus lente que celle des dispositifs électroniques).
- Manque d'interaction entre l'utilisateur et le travail pendant l'exécution
- Les tâches inachevées sont abandonnées.

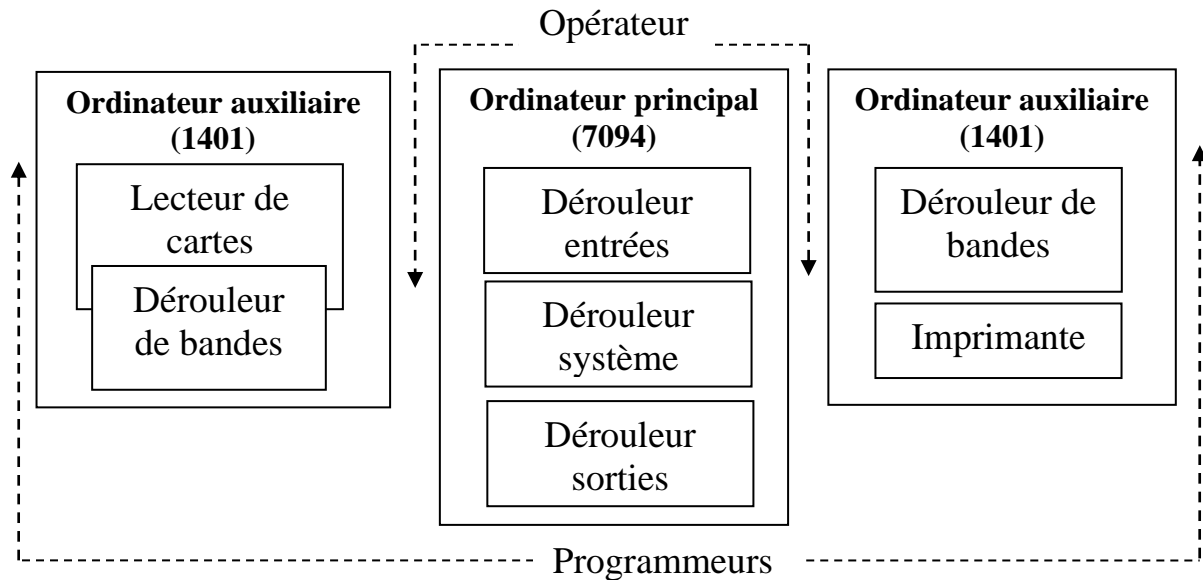


Figure 1.3 : Un système de traitement par lots

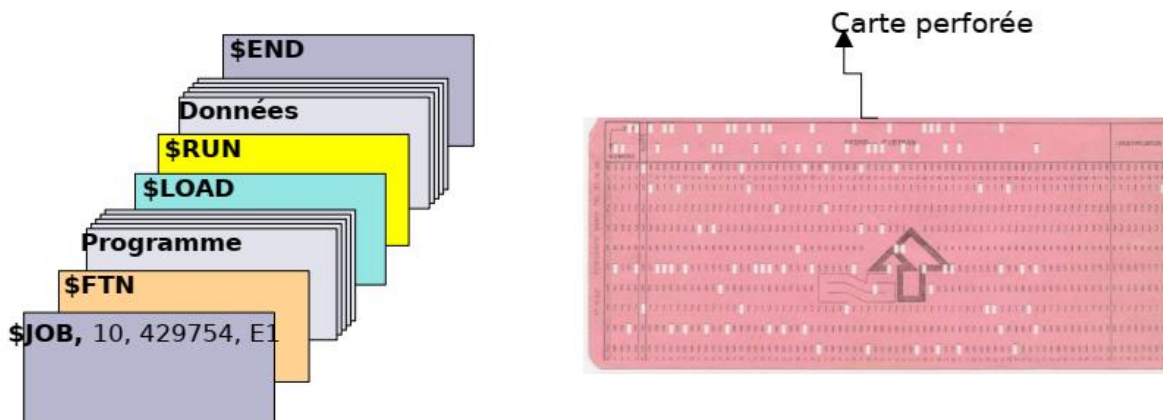


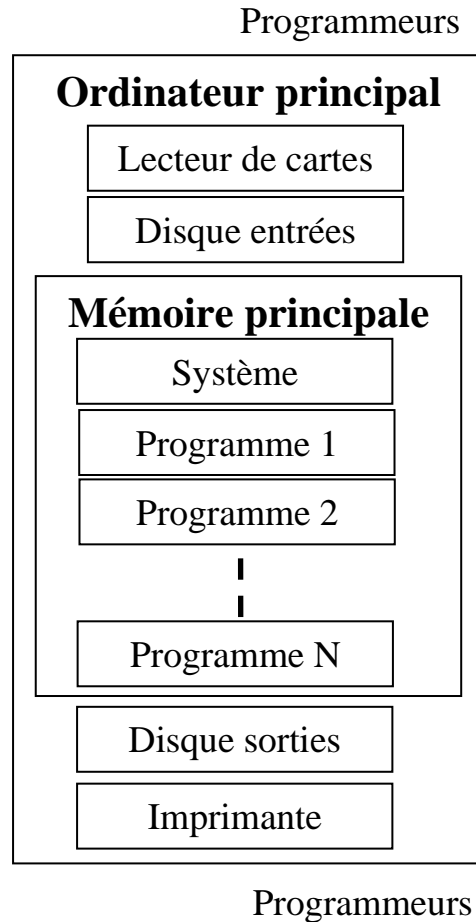
Figure 1.4 : Structure d'un travail FMS typique

### I.7.3. Systèmes multiprogrammes

Les années 1965-1980 ont connu l'utilisation des Machines à circuits intégrés qui sont moins coûteuses et plus performantes ainsi que l'apparition de famille d'ordinateurs compatibles et système d'exploitation unique (machines IBM 360, 370... avec même architecture et jeu d'instructions), qui utilisent des entrées/sorties spoolées (Spool : Simultaneous Peripheral Operation On Line) de façon permettant l'utilisation de la mémoire secondaire au lieu des bandes magnétiques et aussi l'utilisation d'un seul ordinateur pour le calcul et l'échange des entrées/sorties au moyen d'un canal.

Un système multiprogrammé maintient plusieurs travaux en mémoire à la fois (voir figure 1.5). L'idée était alors, pour pallier aux inconvénients du traitement par lots, de maintenir en mémoire plusieurs travaux ou jobs prêts à s'exécuter, et partager efficacement les ressources de la machine entre ces jobs.



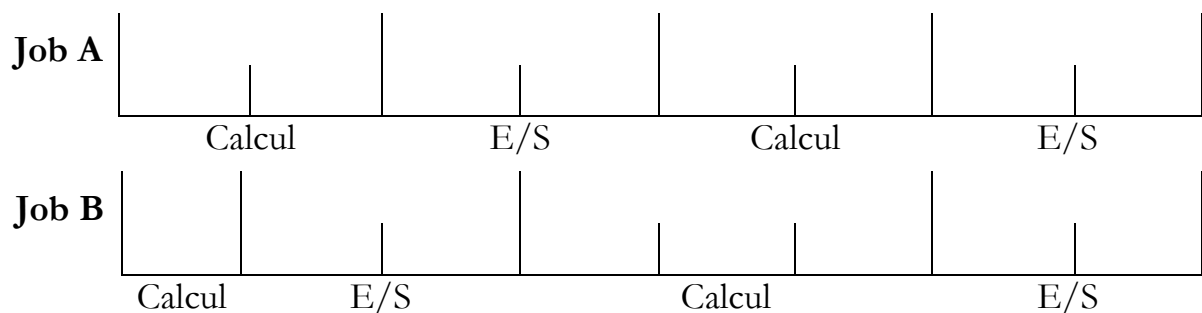


**Figure 1.5 :** Un système de multiprogrammation

En effet, le processeur est alloué à un job, et dès que celui-ci effectue une demande d'E/S, le processeur est alloué à un autre job, éliminant ainsi les temps d'attente de l'unité de traitement chargé des E/S, appelé canal d'E/S.

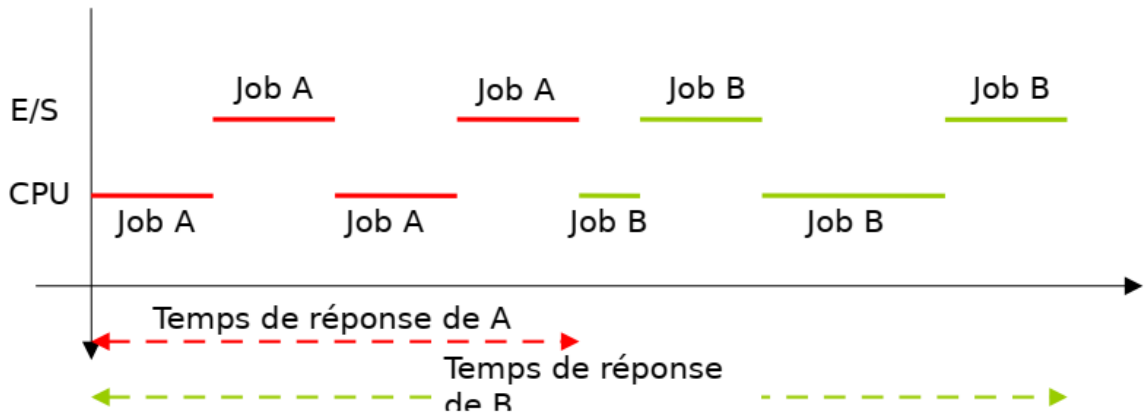
**Exemple**

Soient les deux programmes A et B suivants :

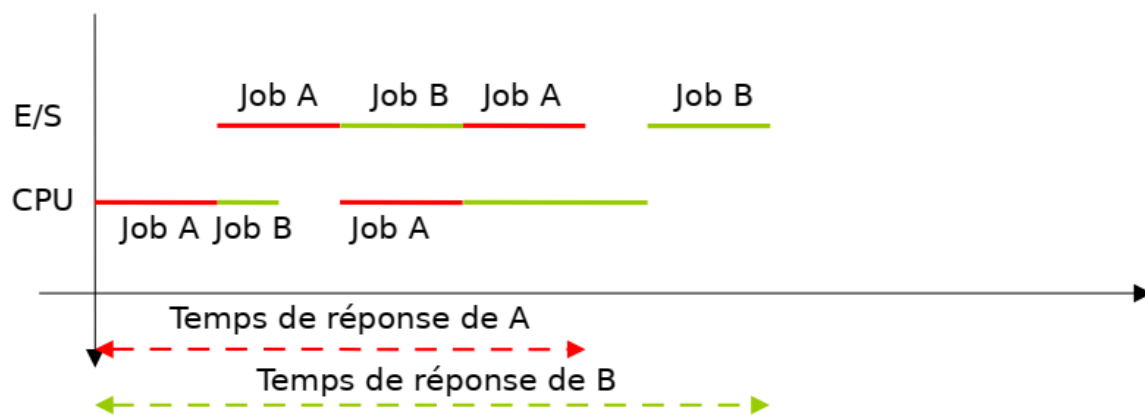


On suppose qu'on a un seul périphérique d'E/S.

**Système mono-programmé**



### Système multiprogrammé



### Comparaison de la Monoprogrammation et de la Multiprogrammation

Monoprogrammation	Multiprogrammation
<ul style="list-style-type: none"> <li>✓ Mauvaise utilisation des ressources (processeur, mémoire, E/S, ...etc.).</li> <li>✓ Temps de réponse imposé par les jobs très longs.</li> <li>✓ S.E. simple ; seule contrainte : protéger la partie résidente du système des utilisateurs.</li> </ul>	<ul style="list-style-type: none"> <li>✓ Possibilité de mieux équilibrer la charge des ressources.</li> <li>✓ Mieux utiliser la mémoire (minimiser l'espace libre).</li> <li>✓ Possibilité d'améliorer le temps de réponse pour les travaux courts.</li> <li>✓ Protéger les programmes utilisateurs des actions des autres utilisateurs, et protéger aussi la partie résidente des usagers</li> </ul>

### I.7.4. Temps partagé (Time Sharing, 1970)

Le temps partagé est une extension logique de la multiprogrammation. C'est une variante du mode multiprogrammé où le temps CPU est distribué en petites tranches appelées **quantum de temps**.

Les systèmes à temps partagé consistent à partager l'unité centrale entre plusieurs tâches par quantum de temps pour assurer leur traitement "simultané" et permettre ainsi à plusieurs utilisateurs connectés en ligne d'être servis à la fois de façon interactive.

L'objectif est d'offrir aux usagers une interaction directe avec la machine par l'intermédiaire de terminaux de conversation, et de leur allouer le processeur successivement durant un quantum de temps, chaque utilisateur aura l'impression de disposer de la machine à lui tout seul. Il peut aussi contrôler le job qu'il a soumis directement à partir du terminal (corriger les erreurs, recompiler, resoumettre le job, ...).

Parmi lesquels, figurent CTSS (Compatible Time Sharing System), MULTICS (MULTiplexed Information and Computing Service) qui est un ancêtre d'Unix et Unix qui est le plus porté. En fait, la plupart des systèmes d'aujourd'hui sont en temps partagé.

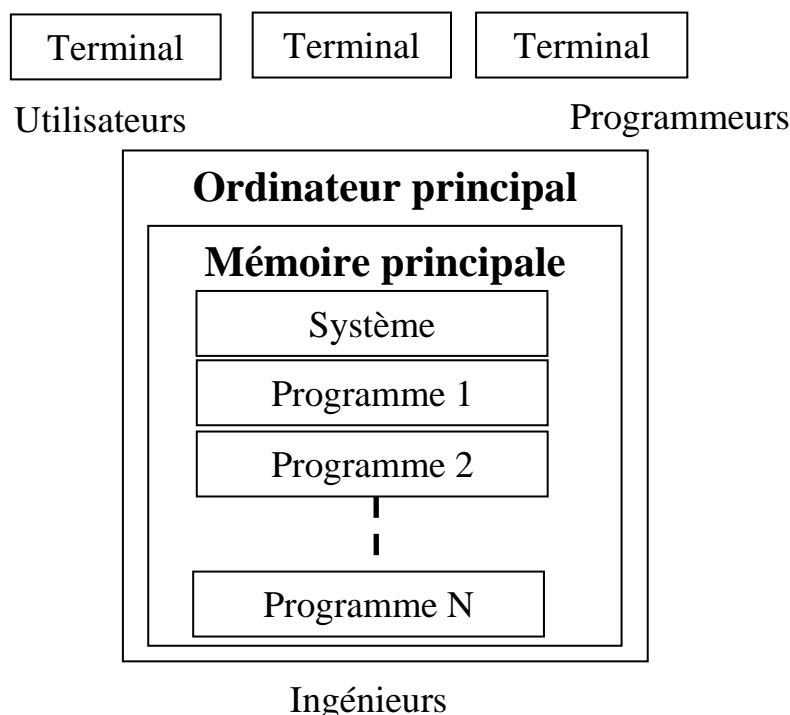


Figure 1.6 : Un système à temps partagé

## I.8. Taxonomie des systèmes d'exploitation

### I.8.1. Systèmes des ordinateurs personnels

Comme le coût du matériel a baissé, il est devenu possible d'avoir un système informatique dédié à un seul utilisateur. Ces types de systèmes informatiques sont appelés ordinateurs personnels ou PC (Personal Computer) ou micro-ordinateurs. Les PC sont apparus dans les années 70.

Les objectifs de ces systèmes (qui n'étaient ni multiutilisateurs, ni multitâches) sont concentrés sur la commodité de l'utilisateur et la rapidité de réaction, la convivialité et l'interactivité avec l'utilisateur, plutôt que de rendre maximale l'utilisation de l'unité centrale et ses périphériques.

Un système informatique interactif (ou assisté) permet la communication directe entre l'utilisateur et le système : l'utilisateur donne des instructions ou commandes au S.E., par clavier ou souris, et attend des résultats immédiats.

Exemples de systèmes : MS-DOS de Microsoft, Macintosh d'Apple.

### I.8.2. Systèmes Parallèles (Parallel Systems)

Les systèmes parallèles, appelés aussi systèmes **fortement couplés (Tightly Coupled)**, possèdent plus d'un processeur en étroite communication, qui partagent le bus de l'ordinateur, l'horloge et parfois la mémoire et les périphériques.

L'avantage de tels systèmes est d'augmenter la **capacité de traitement**, où plusieurs processeurs collaborent pour réaliser une tâche et aussi la **fiabilité** : dans le cas où un des processeurs tombe en panne, le système global ne s'arrête pas, mais ralentit seulement ses travaux.

Les systèmes multiprocesseurs les plus courants utilisent maintenant la fonctionnalité de **multitraitement symétrique (SMP, Symmetric MultiProcessing)**, où chaque processeur exécute une copie du S.E. et ces copies communiquent entre elles lorsque nécessaire.

### I.8.3. Systèmes Distribués et Réseaux

Un développement intéressant a débuté vers le milieu des années 1980 : la croissance des réseaux d'ordinateurs personnels fonctionnant sous des **systèmes d'exploitation réseaux** ou des **systèmes d'exploitation distribués**.

Les **systèmes réseaux (Network Operating Systems)**, permettant l'accès des utilisateurs à des services communs, trop coûteux pour être individuels : stockage de grandes quantités d'informations, impression des documents, processeurs de grandes puissances, ...etc. Ces services sont gérés par des systèmes appelés serveurs, auxquels s'adressent les utilisateurs clients qui disposent chacun de leurs machines individuelles. Un ordinateur qui exécute un système d'exploitation réseau agit indépendamment de tous les autres ordinateurs présents sur le réseau, bien qu'il ait connaissance du réseau et soit capable de communiquer avec d'autres ordinateurs connectés.

Les **systèmes distribués (Distributed Systems)** qui consistent en plusieurs ordinateurs pourvus de leur système d'exploitation et reliés entre eux par des canaux de communication. On parle de systèmes répartis ou **faiblement couplés (Loosely Coupled)**. Les S.E. de ces ordinateurs coopèrent et collaborent pour exécuter une application distribuée. Ces systèmes communiquent de manière suffisamment proche pour donner l'illusion qu'il n'y a qu'un S.E. unique qui contrôle le réseau.

### I.8.4. Systèmes Temps Réels

Ce type de système est un S.E. 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 désirés et donne les résultats pour d'éventuelles interventions.

Un tel système est utilisé lorsqu'il y a des exigences de temps de réponse pour le fonctionnement d'un processeur. Un système d'exploitation temps réel possède des contraintes de temps fixes et bien définies. Le traitement doit être effectué dans les contraintes définies, sinon le système tombe en panne.

### Exemple

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

### I.8.5. Systèmes Embarqués (Embedded Systems)

Un système embarqué est un système prévu pour fonctionner sur : des appareils autonomes (ex. électroménager, robot, automobile), ou des appareils de petite taille (ex. PDA, téléphone, carte à puce). Ces systèmes sont caractérisés par une autonomie réduite, un besoin de gestion avancée de l'énergie, et une capacité de fonctionner avec des ressources limitées. Parmi ces systèmes, on trouve **Palm OS** et **Windows CE** pour PDA, **Java Card** pour cartes à puce, et **TinyOS** pour les réseaux de capteurs (**Sensor networks**), ...etc.

### I.8.6. Systèmes Multi-cœurs (Multicore Systems)

Un ordinateur **multi-cœurs (Chip Multi-Processor)**, combine deux ou plusieurs processeurs, appelés **cœurs (Cores)**, sur une seule puce de silicone, appelée **die (Die)**. L'apparition des architectures multi-cœurs nécessite de repenser la conception des systèmes d'exploitation afin de pouvoir en tirer profit du parallélisme offert par ces architectures. Le défi de conception d'un système d'exploitation pour une architecture multi-cœurs est de permettre une exploitation simple, sûre, et efficace des processeurs multi-cœurs.

Même si les systèmes d'exploitation actuels ne supportent pas totalement les architectures multi cœurs, des améliorations ont été apportées à leur gestion de ressources (notamment la gestion de l'ordonnancement et la gestion des caches) pour prendre en compte ces architectures.

## I.9. Classification des systèmes d'exploitation

Les systèmes d'exploitation peuvent être classés selon différents contextes : l'utilisation, les services fournis et l'architecture matérielle.

### I.9.1. Classement des systèmes selon les contraintes d'utilisation

Selon les contraintes d'utilisation, on distingue :

- Les systèmes mono-utilisateur/mono-tâche où un seul utilisateur utilise le système à la fois et une seule tâche peut être exécutée à la fois. C'est le cas de MS-DOS à titre d'exemple.
- Les systèmes mono-utilisateur/multitâches où un seul utilisateur à la fois peut exécuter plusieurs tâches simultanément comme il est le cas de Windows.

- Les systèmes multi-utilisateurs/multitâches où plusieurs utilisateurs à la fois peuvent exécuter chacun plusieurs tâches simultanément et partager les mêmes ressources matérielles comme c'est le cas d'Unix.

### I.9.2. Classement des systèmes selon les services

Selon les services fournis, on distingue deux catégories :

- Les systèmes temps réel qui sont utilisés dans des domaines spécifiques (procédés, robotique, centrales nucléaires...). Ces systèmes assurent un temps de réponse des tâches critiques court et sont fiables et tolérants aux pannes.
- Les systèmes transactionnels qui sont dédiés à la gestion des bases de données énormes (systèmes de réservation, systèmes bancaires...) et qui permettent de garantir des mises à jour sans incohérence.

### I.9.3. Classement des systèmes selon les architectures

Selon leurs architectures matérielles, les systèmes d'exploitation sont classés en deux catégories :

- Les systèmes mono-processeur qui réfèrent aux systèmes multiprogrammés et à temps partagé. Ces systèmes n'ont besoin que d'un seul processeur qui peut faire des calculs pseudo- parallèles pour faire progresser les tâches à la fois.
- Les systèmes parallèles (multiprocesseurs) comme SunOS 4, SunOS 5, Solaris 2 et Linux qui sont capables d'effectuer un traitement parallèle par plusieurs processeurs. Ils sont fiables et caractérisés par une grande capacité de traitement et un temps de réponse court.

### I.10. Mode d'exécution et appels systèmes

Dans les systèmes d'exploitation, l'exécution est effectuée selon deux modes :

- **Le mode utilisateur** : c'est un mode non protégé réservé à l'exécution des programmes des utilisateurs. Il permet à l'utilisateur de modifier des données de son programme.
- **Le mode noyau** : ce mode protégé est réservé au système d'exploitation. Il permet l'accès au code et données utilisées par ce dernier ainsi que la lecture et l'écriture sur les ports d'entrées/sorties. Aussi, prend-il la charge de protéger les données sensibles.

Un appel système est une interface entre les programmes utilisateurs et le système d'exploitation. Lors de l'appel d'une fonction système telle que fopen, une interruption logicielle Trap survient et le système bascule du mode utilisateur où le programme appelant s'exécutait au mode noyau où l'appel système fopen sera exécutée (voir figure 1.7).

### I.11. Exemples de systèmes d'exploitation :

Plusieurs variantes de systèmes d'exploitation ont apparu dès les années 50 jusqu'aujourd'hui, les fameux sont :

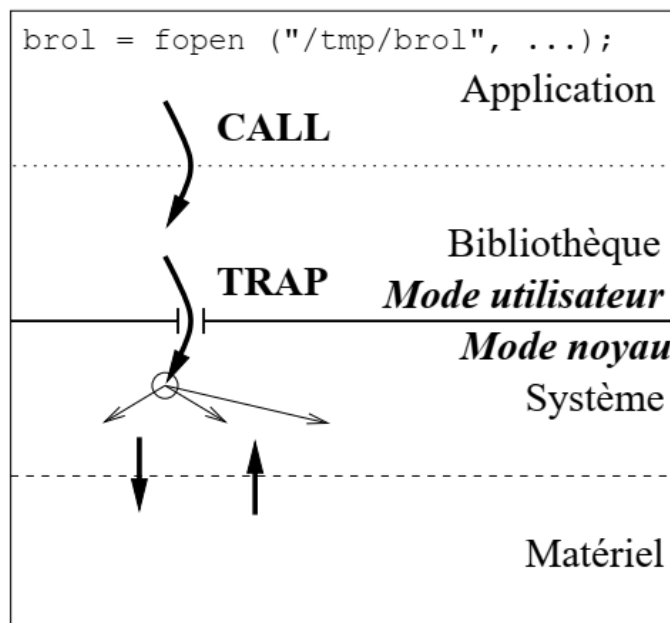


Figure 1.7 : Mécanisme d'appel système

### I.11.1. MS-DOS: Microsoft Disk Operating system

C'est un système mono-utilisateur, il est connu au début des années 80, il est destiné au microordinateur (les premiers IBM-PC 8088), conçu par Bill Gates, associé aujourd'hui avec Windows.

### I.11.2. Le système Windows

Système conçu par Microsoft avec une interface graphique, son nom désigne l'utilisation d'une fenêtre pour chaque exécution. Il est destiné à plusieurs architectures (PC, Station de travail, Portable, Réseau client/serveur, ...).

Il est passé par plusieurs versions comme pour PC : 1.0, 2.0, 3.10, 3.11, 9x, 2000, Me, XP, Vista, Seven, 10, aussi : Windows Server, NT pour l'architecture multi-utilisateur et client/serveur.

### I.11.3. Le système UNIX : Uniplexed Information and Computer Service

Une famille de systèmes en temps partagé proposée pour la plupart des architectures, utilisée pour certains milieux d'industrie et de recherche.

#### Historique

- 1969 : Naissance d'UNICS aux laboratoires de Bell par Ken Thompson → une altération en UNIX.
- 1974 : Publication d'UNIX en langage C par Thompson et Ritchie.
- 1979 : Apparition des versions BSD (Berkeley Software Distribution) de l'université de Berkeley.
- Jusqu'à 1980 : UNIX est offert aux maisons d'enseignements et aux entreprises.
- Aux années 80 : La société AT & T commercialisa les versions System III, System IV et System V

- 1981-1994 : Apparition de Linux.

### ***Les raisons de succès d'UNIX***

- Ecrit à l'aide d'un langage de haut niveau par rapport aux autres systèmes à l'époque.
- Interface utilisateur adaptée.
- Open source → possibilité d'amélioration dans le code source par n'importe qui ce qui résulte l'apparition de plusieurs variantes par plusieurs sociétés comme : System V de AT & T, XENIX par Microsoft, ULTRIX par DEC, pour Linux → Redhat, Coldera, ...
- Disponibilité de versions gratuites.

### ***Les services offerts par le noyau UNIX***

- La gestion des événements matériels.
- La gestion des ressources.
- La gestion des structures de données.
- La gestion de la mémoire et le processeur.
- Fournit des services élémentaires.

C'est un système multitâche, multi-utilisateurs, multisections, et multipostes.