

**Université Mohamed Kheider Biskra**  
**Faculté des sciences exactes et sciences de la nature et de la vie**  
**Département d'informatique**

**Support de cours**

**Module : Systèmes d'exploitation I**

**Chapitre I : Introduction aux systèmes d'exploitation**

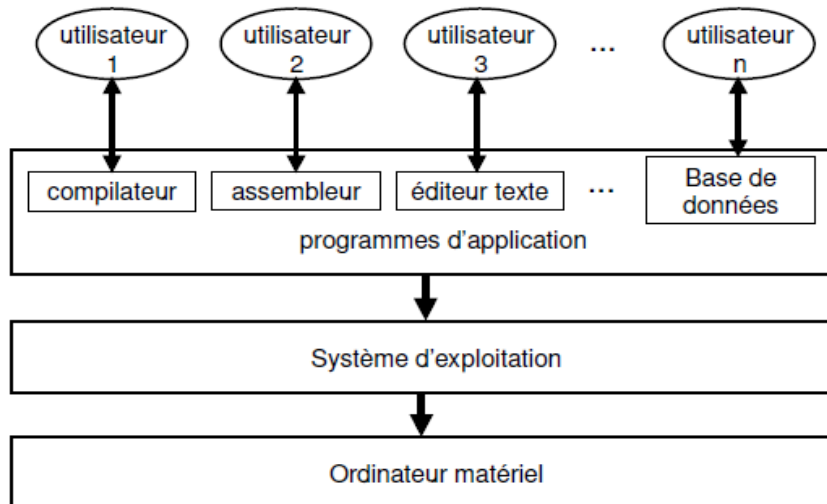
**Niveau : 2LMD**

**Chargé de module : Mme. D. Boukhrouf**

**Année universitaire 2017/2018**

## 1. Qu'est-ce qu'un système d'exploitation

Un système d'exploitation est un programme qui doit permettre aux utilisateurs d'utiliser les fonctionnalités d'un ordinateur. Mais il doit aussi aider le programmeur à développer des logiciels de la façon la plus efficace possible. Un système d'exploitation est mis en route dès qu'on met en marche l'ordinateur. Dans les grosses machines, il est toujours en exécution. Le système constitue donc une interface entre l'utilisateur et la machine physique.



Le **système d'exploitation** 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 but d'un système d'exploitation consiste donc à développer des applications sans se soucier des détails de fonctionnement et de gestion du matériel. Ainsi, par exemple, on peut effectuer la lecture d'un fichier par un simple appel système : `read(fd, buf, 255)`; et peu importe que le fichier en question se trouve dans un disque magnétique, un DVD ou un disque en RAM.

## 2. Évolution des systèmes d'exploitation

L'apparition et l'évolution des systèmes d'exploitation est lié à l'évolution des ordinateurs.

### 2.1 Première génération (1945-1955)

Dans les années 40, les premiers « ordinateur » apparaissent, les programmes et les données sont codés directement en binaire sur des tableaux d'interrupteurs, puis plus tard (1950) sur des cartes perforées (la présence ou l'absence de trous correspondent à 0 ou 1), ils sont chargés en mémoire, exécutés et mis au point à partir d'un pupitre de commande. C'est la même équipe qui conçoit, construit, programme, administre et maintient la machine.

L'arrivée de l'assembleur facilite l'écriture des programmes, mais l'utilisation des machines est toujours la même. L'ordinateur ne traite qu'un programme à la fois. Il faut pour chaque travail (job), en amont insérer les cartes dans un lecteur de cartes, en aval imprimer les résultats, ce sont des opérations très lentes par rapport au travail du CPU. Par la suite, c'est un opérateur qui se charge de l'exploitation de la machine mais le principe reste le même.

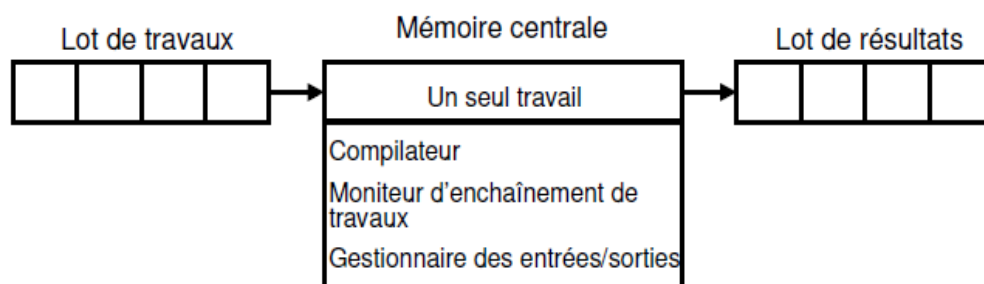
### 2.2 Deuxième génération (1955-1965)

Avec l'apparition du transistor (1955 environ), les ordinateurs sont beaucoup plus fiables, mais coûtent très cher (seules de grandes entreprises privées ou publiques en possèdent). Ce sont maintenant des équipes différentes qui conçoivent, construisent, programment, administrent et maintiennent la machine.

Pour chaque travail (job) à réaliser, le programmeur transmet son paquet de cartes perforées (programmes en assembleur ou en FORTRAN) à l'opérateur qui les soumet au compilateur puis à l'ordinateur. Une fois l'exécution terminée, l'opérateur récupère le résultat sur l'imprimante. Comme les ordinateurs sont très coûteux, on cherche à réduire les temps d'attente en automatisant les opérations manuelles.

Les gros ordinateurs disposent alors de trois dérouleurs de bande : un pour le système d'exploitation, un pour le programme et ses données, un pour les données en sortie. En annexe, il existe des machines pour transcrire les cartes perforées (apportés par les programmeurs) sur bandes et des machines pour imprimer les données en sortie (contenues sur la bande) sur papier. On parle d'impression off-line (l'imprimante n'est pas directement connectée à l'ordinateur). Mais le calculateur principal ne fonctionne pas tout le temps : beaucoup de temps est encore perdu lors des déplacements des opérateurs qui sont chargés d'alimenter les machines en cartes, papier, bandes.

On procède au traitement par lots (batch) : plusieurs travaux sont transcrits sur une même bande d'entrée. Le calculateur principal lit le 1er job, puis lorsque celui-ci est terminé, lit le 2nd, etc... jusqu'à la fin de la bande. Un interprète de commande permet le chargement du programme et des données puis l'exécution du programme, le moniteur est le programme chargé du séquençage des travaux des utilisateurs et de la continuité des opérations. Le moniteur des années 60 est le précurseur du système d'exploitation.



### 2.3 Troisième génération (1965-1980)

Au milieu des années 60, l'apparition des circuits intégrés permet une grande avancée sur la performance et le coût des ordinateurs. Des familles de machines partageant le même langage machine, le même système d'exploitation se mettent en place.

Les travaux sont entièrement traités par l'ordinateur sans passer par des machines annexes, les jobs sont stockés sur des fichiers disques. Comme l'accès sur un disque est aléatoire (dans le sens de non-séquentiel) le moniteur peut choisir l'ordre des travaux.

Dans le moniteur, le planificateur (scheduler) assure cette tâche. Mais les moniteurs batch exécutent toujours un seul job à la fois, à tout instant un seul programme se trouve en mémoire et peut seul exploiter le CPU.

On arrive à la multiprogrammation. La mémoire est partagée entre différents programmes en attente d'une donnée en entrée par exemple peut être suspendu temporairement au profit d'une tâche. Le but étant d'exploiter au maximum le CPU. On appelle cette technique le spouleur (de SPOOL : Simultaneous Peripheral Operation On Line). Dans le système d'exploitation, l'allocateur (dispatcher) s'occupe de la gestion instantanée du CPU en tenant compte de la planification établie par le scheduler. Il a fallu mettre en place des systèmes de contrôle des accès mémoires, de protection des données.

## Chapitre1 : Introduction aux systèmes d'exploitation

---

On voit apparaître également le temps-partagé (time-sharing) pour des ordinateurs multi-utilisateurs (un ordinateur central connecté à plusieurs terminaux).

### 2.4 Quatrième génération (1980-Maintenant)

Depuis les années 80, on assiste au développement des ordinateurs personnels, à l'amélioration constante du taux d'intégration des circuits (LSI : Large Scale Integration, VLSI, Very LSI), à la baisse des coûts et au développement des réseaux de communications, au développement des réseaux locaux, à l'explosion d'internet. Les interfaces Homme/Machine sont toujours améliorées.

### 3. Les principales fonctions d'un système d'exploitation

Les rôles du système d'exploitation sont divers :

- **Gestion du processeur** : le système d'exploitation est chargé de gérer l'allocation du processeur entre les différents programmes grâce à un **algorithme d'ordonnancement**. Le type d'ordonnanceur est totalement dépendant du système d'exploitation, en fonction de l'objectif visé.
- **Gestion de la mémoire vive** : le système d'exploitation est chargé de gérer l'espace mémoire alloué à chaque application et, le cas échéant, à chaque usager. 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. En contrepartie cette mémoire est beaucoup plus lente.
- **Gestion des entrées/sorties** : le système d'exploitation permet d'unifier et 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. Il permet à ce titre de «tuer» une application ne répondant plus correctement.
- **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.
- **Gestion des informations** : le système d'exploitation fournit un certain nombre d'indicateurs permettant de diagnostiquer le bon fonctionnement de la machine.

### 4. Composantes d'un système d'exploitation

Le système d'exploitation est composé d'un ensemble de logiciels permettant de gérer les interactions avec le matériel. Parmi cet ensemble de logiciels on distingue généralement les éléments suivants :

- Le **noyau** (en anglais **kernel**) représentant les fonctions fondamentales du système d'exploitation telles que la gestion de la mémoire, des processus, des fichiers, des entrées-sorties principales, et des fonctionnalités de communication.
- L'**interpréteur de commande** (en anglais **shell**, traduisez «*coquille*» par opposition au noyau) permettant la communication avec le système d'exploitation par l'intermédiaire d'un langage de commandes, afin de permettre à l'utilisateur de piloter les périphériques en ignorant tout des caractéristiques du matériel qu'il utilise, de la gestion des adresses physiques, etc.

· Le **système de fichiers** (en anglais «*file system*», noté *FS*), permettant d'enregistrer les fichiers dans une arborescence.

### **5. Les différents types d'un SE**

#### **5.1. Systèmes multitâches**

Un système d'exploitation est dit «**multi-tâche**» (en anglais *multithreaded*) lorsque plusieurs «**tâches**» (également appelées *processus*) peuvent être exécutées simultanément.

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.

Un système est dit **préemptif** lorsqu'il possède un **ordonnanceur** (aussi appelé *planificateur*), qui répartit, selon des critères de priorité, le temps machine entre les différents processus qui en font la demande.

Le système est dit à **temps partagé** lorsqu'un quota de temps est alloué à chaque processus par l'ordonnanceur. C'est notamment le cas des systèmes multi-utilisateurs qui permettent à plusieurs utilisateurs d'utiliser simultanément sur une même machine des applications différentes ou bien similaires : le système est alors dit «**système transactionnel**». Pour ce faire, le système alloue à chaque utilisateur une tranche de temps.

#### **5.2. Systèmes multi-processeurs**

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 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.

#### **5.3. 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 / Window Smartphone

#### **5.4. Systèmes temps réel**

Les **systèmes temps réel** (*real time systems*), essentiellement utilisés dans l'industrie, sont des systèmes dont l'objectif est de fonctionner dans un environnement contraint temporellement.

Un système temps réel doit ainsi fonctionner de manière fiable selon des contraintes temporelles spécifiques, c'est-à-dire qu'il doit être capable de délivrer un traitement correct des informations reçues à des intervalles de temps bien définis (réguliers ou non).

Voici quelques exemples de systèmes d'exploitation temps réel :

- OS-9 ;

- RTLinux (RealTime Linux) ;
- QNX ;
- VxWorks.

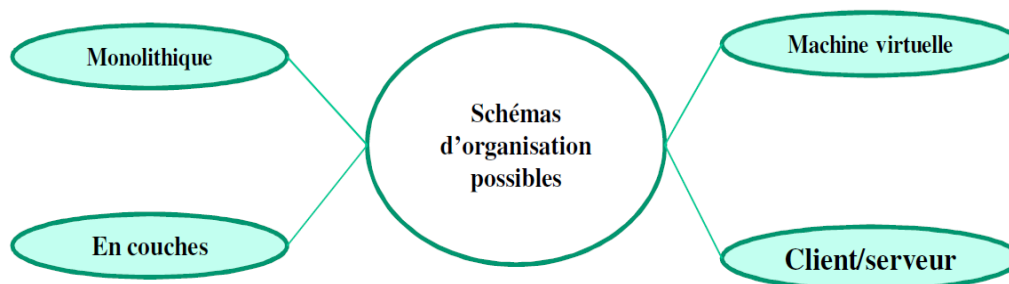
### 5.5. Exemples de systèmes d'exploitation

On distingue plusieurs types de systèmes d'exploitation, selon qu'ils sont capables de gérer simultanément des informations d'une longueur de 16 bits, 32 bits, 64 bits ou plus.

Systeme	Codage	Mono-utilisateur	Multi-utilisateur	Mono-tâche	Multitâche
DOS	16 bits	X		X	
Windows3.1	16/32 bits	X			non préemptif
Windows95/98/Me	32 bits	X			coopératif
WindowsNT/2000	32 bits		X		préemptif
WindowsXP	32/64 bits		X		préemptif
Unix / Linux	32/64 bits		X		préemptif
MAC/OS X	32 bits		X		préemptif
VMS	32 bits		X		préemptif

### 6. Structure interne d'un système d'exploitation

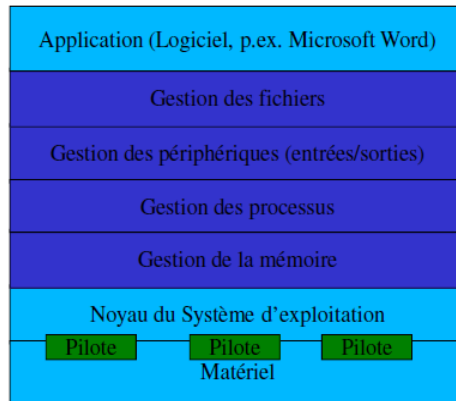
Le système d'exploitation, appelé noyau ou kernel, gère le matériel et fournit aux programmes une interface d'appels système. Les appels système permettent aux programmes de créer et de gérer des processus et des fichiers.



#### 6.1. Structure en couches

Le système d'exploitation est structuré en couches. Chaque couche utilise les fonctions des couches inférieures. La principale difficulté est la définition des différentes couches. Par exemple, on peut l'organiser en cinq couches, comme montré sur la figure suivante :

- Au plus bas niveau on trouve le noyau, l'interface entre le matériel et le logiciel. Il se charge, en utilisant les fonctions fournies par le matériel, de gérer la UCT, les interruptions et les processus (la communication et la synchronisation). Il doit entièrement résider en mémoire.
- Au second niveau, on trouve le gestionnaire de la mémoire qui se charge du partage de la mémoire entre les processus en attente d'exécution.

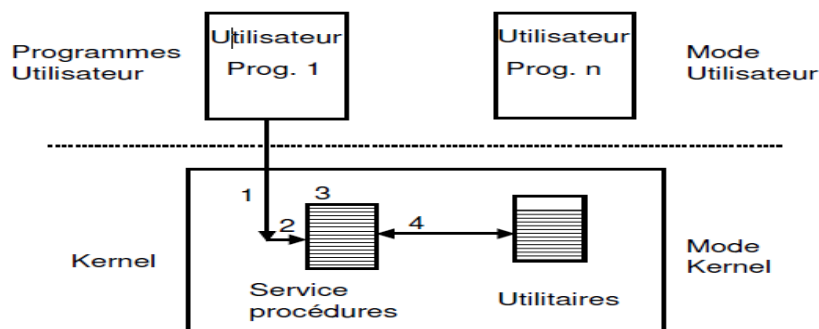


**Figure 2. Structure en couches**

- Au troisième niveau, on a le module de gestion des entrées/sorties qui se charge de gérer tous les périphériques (clavier, écran, disques, imprimantes, etc.).
- Au quatrième niveau, on trouve 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.
- Au cinquième niveau, on a 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 créer de nouveaux processus et leur attribuer un niveau de priorité : 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.

## 6.2. Structure monolithique

Les systèmes d'exploitation sous une structure monolithique sont un ensemble de procédures de — presque — le même niveau : une procédure principale qui appelle la procédure de service requise, des procédures de service qui exécutent les appels système et un ensemble de procédures utilitaires qui assistent les procédures de service, par exemple la recherche de données des programmes utilisateur. Unix et Linux sont des exemples de systèmes monolithiques.

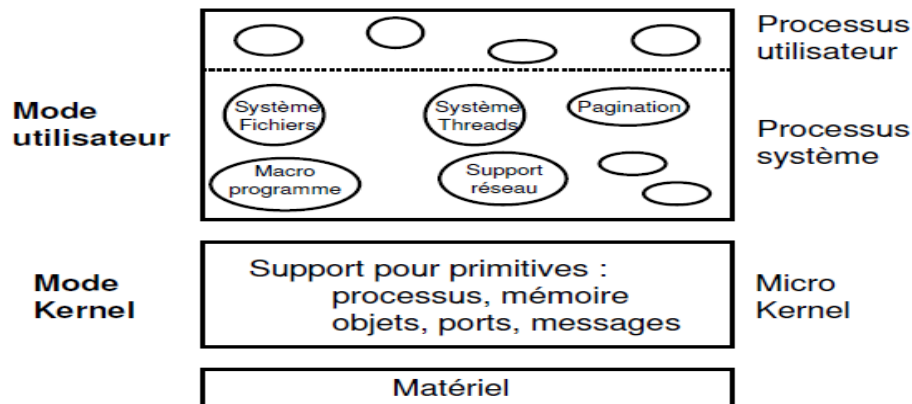


## 6.3. Micro-kernel

Une architecture plus moderne que celle monolithique est l'architecture micro-kernel utilisée en MACH5/HURD6, Minix et NT. L'attribut principal qui distingue les micro-kernels des kernels monolithiques est l'implantation de leurs architectures respectives en mode superviseur (kernel mode) et en mode usager (user mode). L'architecture monolithique met en œuvre tous les services du Système d'exploitation (contrôleurs de dispositifs, mémoire virtuelle, système de fichiers, réseaux, etc) dans le domaine du mode superviseur de l'UCT.

## Chapitre 1 : Introduction aux systèmes d'exploitation

Par contre, l'architecture micro-kernel fait une division entre les services du Système d'exploitation, en les divisant en « haut-niveau » implantés dans le domaine de l'utilisateur et « bas-niveau » implantés dans l'espace du mode superviseur.



### 6.4. Le modèle client/serveur

Dans le modèle client/serveur Le système d'exploitation est composé d'un noyau et d'un ensemble de serveurs. Le noyau gère la communication entre les clients et les serveurs. Les clients sont les demandeurs de services. Par exemple, pour demander un service, comme la lecture d'un bloc d'un fichier, un processus utilisateur (aussi appelé processus client) envoie une requête à un processus serveur qui effectue le travail et renvoie une réponse. Les serveurs s'exécutent en mode utilisateur. Ils ne peuvent donc pas accéder directement au matériel. Par conséquent, une erreur ou un bogue dans le serveur de fichiers, par exemple, n'affectera pas, en général, l'ensemble de la machine. Les dégâts se limitent au serveur.

### 6.5. Machines virtuelles

Le cœur du système se charge de la multiprogrammation en fournissant à la couche au dessus plusieurs machines virtuelles. Des systèmes d'exploitation différents peuvent tourner au dessus d'une machine virtuelle.

Exemples de machines virtuelles :

- IBM VM : offre à chaque usager sa propre machine virtuelle monotâche. Les machines virtuelles étaient planifiées avec du temps partagé.
- Java : les programmes compilés en Java tournent sur une machine virtuelle (JVM).
- VMWare7: sur PC, exécute en même temps des sessions Windows, Linux, OS/2, etc.
- Nachos : SE qui s'exécute dans une machine virtuelle MIPS, qui tourne sur Unix.