

# Chapitre 1 : Introduction aux Systèmes d'exploitation

**Dr. Mandicou BA**

[mandicou.ba@esp.sn](mailto:mandicou.ba@esp.sn)

<http://www.mandicouba.net>

Diplôme Universitaire de Technologie (DUT, 2<sup>e</sup> année)  
Diplôme Supérieure de Technologie (DST, 2<sup>e</sup> année)  
en Informatique



ÉCOLE SUPÉRIEURE POLYTECHNIQUE

[www.esp.sn](http://www.esp.sn)



# Plan du Chapitre

- 1 Pourquoi et qu'est-ce qu'un système d'exploitation?
- 2 Évolution des systèmes d'exploration
- 3 Positionnement d'un Système d'exploitation
- 4 Fonctions d'un système d'exploitation
- 5 Exemple de systèmes d'exploitation
- 6 Classification des systèmes d'exploitation

# Sommaire

- 1 Pourquoi et qu'est-ce qu'un système d'exploitation?
- 2 Évolution des systèmes d'exploration
- 3 Positionnement d'un Système d'exploitation
- 4 Fonctions d'un système d'exploitation
- 5 Exemple de systèmes d'exploitation
- 6 Classification des systèmes d'exploitation

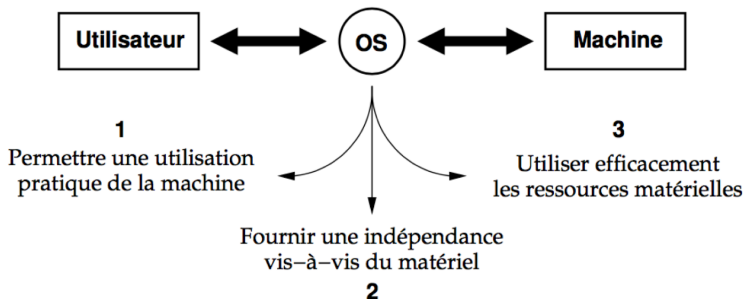
# Pourquoi un système d'exploitation

- ☞ Un système informatique est constitué :
  - ▢ d'un ou de plusieurs processeurs
  - ▢ d'une mémoire principale et des mémoires secondaires
  - ▢ d'un clavier, d'un écran, d'une interface réseau, etc.
  - ▢ d'une imprimante, d'une carte TV
  - ▢ etc.
- ☞ C'est donc un système très complexe.
- ☞ Astuce : trouver un moyen de gérer tout ce matériel et offrir une interface simplifiée à ce programme pour qu'il manipule tout ce matériel facilement et d'une manière optimale
  - ▢ **Logiciels systèmes** : le plus important de ces logiciels est le **Système d'exploitation**

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

## Définition 1

- ☛ Programme qui permet d'exploiter les ressources matérielles
- ☛ fournit au programmeur d'application un environnement avec facilité d'emploi et utilisation efficace des ressources.



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

## Définition 2

- ☛ Un « **Système d'Exploitation** » (SE ou OS pour Operating System) est le programme fondamental des programmes systèmes.
- ☛ Son rôle est de *gérer* tous les périphériques et de fournir aux programmeur utilisateur une interface simplifiée avec le matériel
- ☛ Il contrôle donc toutes les ressources de l'ordinateur et fournit la base sur laquelle seront construites les programmes d'application

# Facilité d'emploi

- ☛ Le système d'exploitation cache les détails matériels sous une couche d'abstraction.
- ☛ Il fournit une API (Application Programming Interface) pour utiliser les ressources.

## Exemple 1

- *Programme qui permet d'exploiter les ressources matérielles*
- *Fournit au programmeur d'application un environnement avec facilité d'emploi et utilisation efficace des ressources.*

# Utilisation de ressources

- ☛ Le système d'exploitation est un gestionnaire de ressources ; il gère l'allocation :
  - des processeurs,
  - de la mémoire,
  - des périphériques
  - etc.

## Attention 1

- *Ne pas confondre système d'exploitation et interface utilisateur !*



# Sommaire

- 1 Pourquoi et qu'est-ce qu'un système d'exploitation?
- 2 Évolution des systèmes d'exploration**
- 3 Positionnement d'un Système d'exploitation
- 4 Fonctions d'un système d'exploitation
- 5 Exemple de systèmes d'exploitation
- 6 Classification des systèmes d'exploitation

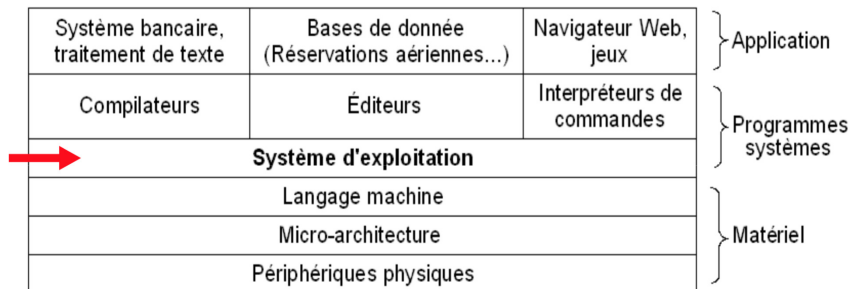
# Évolution des Systèmes d'exploration

- ❶ La première génération (1954-1955) : **tubes à vides**
- ❷ La deuxième génération (1955-1965) : **transistors et traitement par lots**
- ❸ La troisième génération (1965-1980) : **circuits intégrés et multiprogrammation**
- ❹ La quatrième génération (1980-1990) : **les ordinateurs personnels**
- ❺ La cinquième génération (1990-aujourd'hui) : **les ordinateurs personnels portables et de poche**

# Sommaire

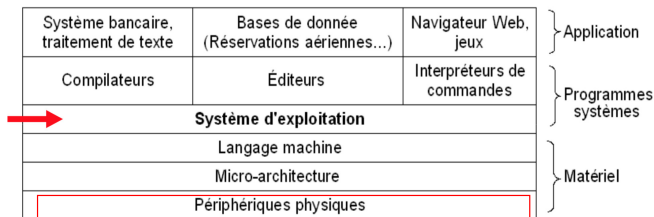
- 1 Pourquoi et qu'est-ce qu'un système d'exploitation?
- 2 Évolution des systèmes d'exploration
- 3 Positionnement d'un Système d'exploitation**
- 4 Fonctions d'un système d'exploitation
- 5 Exemple de systèmes d'exploitation
- 6 Classification des systèmes d'exploitation

# Structuration d'un ordinateur



Positionnement du système d'exploitation dans la couche des programmes  
qui contribuent à l'utilisation d'un ordinateur

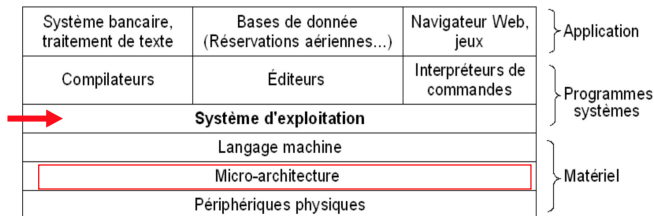
# Structuration d'un ordinateur



Positionnement du système d'exploitation dans la couche des programmes  
qui contribuent à l'utilisation d'un ordinateur

- ☛ **Les périphériques physiques** : font partie de la couche « matériel ». Elles intéressent plus les ingénieurs électroniciens car elles sont composées des circuits intégrés, câbles, alimentation, etc.

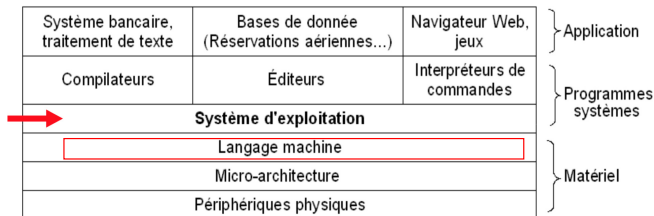
# Structuration d'un ordinateur



Positionnement du système d'exploitation dans la couche des programmes qui contribuent à l'utilisation d'un ordinateur

- Micro-Architecture** : correspond aux *programmes de contrôle de périphériques* (interprète). A chaque tour d'horloge, extraire des **registres** un ou deux opérandes , et, faire une opération logique sur eux, puis, le résultat est stocké sur un ou plusieurs registres.

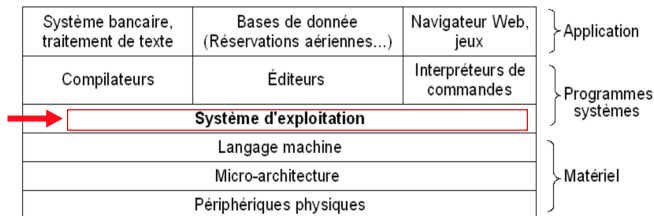
# Structuration d'un ordinateur



Positionnement du système d'exploitation dans la couche des programmes qui contribuent à l'utilisation d'un ordinateur

- ☛ **Langage machine** : Correspond au programme (en assembleur) qui permet d'effectuer des instructions élémentaires (ADD, JUMP, MOVE, etc.) pour déplacer les données en mémoire et faire des calculs ou comparaison entre les valeurs des registres

# Structuration d'un ordinateur

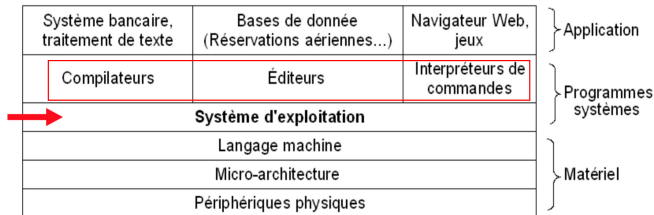


Positionnement du système d'exploitation dans la couche des programmes  
qui contribuent à l'utilisation d'un ordinateur

- ☛ **Système d'exploitation** : c'est une couche logiciel qui **cache partiellement le matériel** et fournit au développeur **un jeu d'instructions plus pratique**, qui englobe souvent plusieurs instructions du langages machines



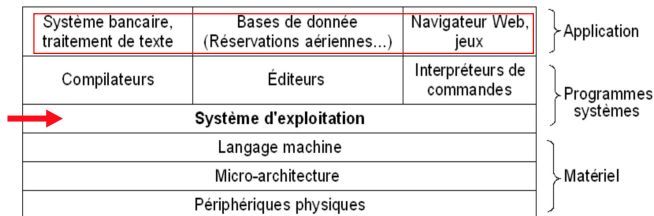
# Structuration d'un ordinateur



Positionnement du système d'exploitation dans la couche des programmes  
qui contribuent à l'utilisation d'un ordinateur

- Les autres programmes systèmes** : la différences entre ces logiciels et le SE est que le SE fonctionne en mode noyau (mode super-utilisateur) alors que les autres fonctionnent en mode utilisateur. On peut par exemple changer de compilateur mais pas le gestionnaire d'interruptions d'horloge.

# Structuration d'un ordinateur



Positionnement du système d'exploitation dans la couche des programmes qui contribuent à l'utilisation d'un ordinateur

- ☛ **Application** : on y trouve tous les programmes (acheté ou non, conçus, etc.) qui répondent aux besoins particuliers des utilisateurs.

# Sommaire

- 1 Pourquoi et qu'est-ce qu'un système d'exploitation?
- 2 Évolution des systèmes d'exploration
- 3 Positionnement d'un Système d'exploitation
- 4 Fonctions d'un système d'exploitation**
- 5 Exemple de systèmes d'exploitation
- 6 Classification des systèmes d'exploitation

# Fonctions d'un Système d'exploitation

Un système d'exploitation assure globalement deux fonctions essentielles :

- ❶ **Extension de la machine**
  - Machine virtuelle ou machine étendue
- ❷ **Gestion des ressources**

# Extension de la machine

- ☛ Fournir à l'utilisateur une l'équivalent d'une machine virtuelle plus facile à programmer que la machine réelle.
- ☛ Masquer les éléments fastidieux liés au matériel (gestion des interruption, de la mémoire, des horloges, des périphériques)

## Exemple 2 (Écrire sur le disque dur)

- *Il faut manipuler environ 20 commandes : lecture/écriture de données, adresse du bloc à lire, nombre se secteur par piste, déplacement du bras, formatage des pistes, remise à zero et re-calibrage*
- ☛ Le SE créé une interface de programmation plus **abstraite**, qui lui permet de manipuler les périphériques d'une manière plus simple.

## Exemple 3 (Encore le disque dur)

- *Présenter le disque dur comme un fichier qu'il faut ouvrir, manipuler et fermer*

# Extension de la machine

- ☛ L'interface de programmation offerte par le SE est un ensemble de services que les programmeurs peuvent solliciter par le billet d'instruction spéciales appelées « appels systèmes ».

# Gestions des ressources

- ☛ Un ordinateur est composé d'un ensembles de ressources
- ☛ Plusieurs programmes peuvent s'exécuter sur cet ordinateur.
- ☛ Ces programmes nécessitent de la mémoire, des accès disque et des accès ou utilisation de ressources
  - ➡ Le rôle du SE des de gérer de manière **équitable**, **optimale** et **sans conflit** l'allocation des processeurs, de la mémoire et des périphériques d'entrée/sorties aux différents programmes concurrents qui les sollicitent.

## Exemple 4

- *Impression simultanée*
- *Accès aux fichiers*
- *Autorisation/Interdiction d'accès à certaines ressources*

# Gestions des ressources

## Partage de ressources

- ☛ **Dans le temps** : gérer dans le temps l'accès à une imprimante, à un fichier ou à la CPU.
  - le choix du programme qui va utiliser la ressource est la tâche du CPU.
- ☛ **Dans l'espace** : Au lieu d'attendre son tour, certaines ressources permettes d'être partagées.
  - On peut partager la mémoire principale entre les processus ou programmes actifs qui s'exécute à un instant  $t$ .

## Remarque 1

- *Le rôle du SE est de trouver la bonne équilibre entre le partage de ressources dans le temps et dans l'espace.*
  - *Ne pas allouer toute la mémoire à un seul programme*
  - *Ne pas partager le mémoire entre tous les programmes en un instant  $t$*



# Gestions des ressources

## Les ressources à gérer

- ☞ Les processus
- ☞ Les fichiers
- ☞ Les entrées/sorties
- ☞ La mémoire
- ☞ etc.

# Sommaire

- 1 Pourquoi et qu'est-ce qu'un système d'exploitation?
- 2 Évolution des systèmes d'exploration
- 3 Positionnement d'un Système d'exploitation
- 4 Fonctions d'un système d'exploitation
- 5 Exemple de systèmes d'exploitation**
- 6 Classification des systèmes d'exploitation

# Liste non exhaustive de systèmes d'exploitation

- ☛ OS/2 de IBM
- ☛ Mac OS : le premier SE des ordinateurs Apple Machintosh, qui succède au système Lisa de Apple II, et a été suivi Mac OS X puis iPhone OS.
- ☛ La famille DOS : dont PC-DOS (version fournie par IBM avec les premiers x86), MS-DOS, FreeDOS, etc.
- ☛ Windows 9x : le SE fenêtré de windows basé sur MS-DOS
- ☛ Windows NT jusqu'au Windows 10
- ☛ Dérivée de UNIX sous différentes déclinaison : BSD, System V, etc. :
  - GNU/Linux : un SE libre s'appuyant sur le noyau Linux et les outils GNU ; Debian, Ubuntu, Mandriva, Red Hat, Fedora, Suse, etc.
  - La famille BSD : un effort réussi pour rendre sa liberté aux système Berkely ; NetBSD, OpenBSD, FreeBSD, etc.
  - UNIX propriétaire : AIX (IBM, System V), A/UX (Apple, System V),...

# Sommaire

- 1 Pourquoi et qu'est-ce qu'un système d'exploitation?
- 2 Évolution des systèmes d'exploration
- 3 Positionnement d'un Système d'exploitation
- 4 Fonctions d'un système d'exploitation
- 5 Exemple de systèmes d'exploitation
- 6 Classification des systèmes d'exploitation**

# Critère de classification

- ☞ Libre et gratuit / Propriétaire et payant
- ☞ Services rendus
- ☞ Architecture de fonctionnement
- ☞ Architecture matérielle supportée
- ☞ Évolution
- ☞ Contraintes de temps réel

# Libre et gratuit / Propriétaire et payant

## ☞ Libre et gratuit

- Libre : on peut avoir le code source
- Gratuit : on peut l'avoir gratuitement

### Exemple 5 (SE Libres et Gratuits)

- *Les distributions GNU/Linux (Debian, Fedora, Suse, etc.), NetBSD sont libre et gratuits*

## ☞ Propriétaire et payant : il faut payer pour pouvoir les utiliser

# Services rendus

- ☛ **Mono-tâche** : le SE ne peut exécuter qu'une seule tâche à la fois .
  - C'est l'exemple du DOS
- ☛ **Multi-tâches** : le SE peut exécuter plusieurs tâches ou processus simultanément :
  - Par exemple, compiler un programme et consulter le fichier source du programme correspondant.
- ☛ **Mono-utilisateur** : Il ne peut y avoir qu'un seul utilisateur qui utilise le système en un instant donné.
  - C'est le cas du DOS, Windows, ...
- ☛ **Multi-utilisateurs** : le SE peut gérer plusieurs utilisateurs au même temps. Le SE partage l'accès aux ressources entre ces utilisateurs.
  - C'est le cas de UNIX.

# Architecture de fonctionnement

## ☛ **Système centralisé :**

- l'ensemble du système est entièrement présent sur la machine considérée.
- les autres machines reliées à celle-ci sont vues comme des entités étrangères disposant elles aussi d'un SE centralisé.
- le SE ne gère que les ressources de la machine sur laquelle il est installé

## ☛ **Système réparti (distribué) :**

- Les différentes abstractions du SE sont réparties sur plusieurs machines, mais apparaît aux utilisateurs comme une seule machine virtuelle.
- L'utilisateur n'a pas à se soucier de la localisation des ressources
- Avantages : exécution parallèle des tâches et tolérance aux pannes.



# Architecture matérielle supportée

## ☛ Mono-processeur :

- La machine a **un seul processeur**  $\mapsto$  **pseudo-parallélisme**
- Illusion de parallélisme dans l'exécution des tâches (processus) car il y a une rapide commutation entre les différents processus qui s'exécutent sur la machine

## ☛ Multi-processeurs :

- La machine possède plusieurs **processeurs**. Chacun s'occupe d'une tâche. Plusieurs architectures sont possibles
  - ➡ **SIMD** (Single Instruction Multiple Data) : tous les processeurs exécutent les mêmes instructions, mais sur des données différentes.
  - ➡ **MIMD** (Multiple Instructions Multiple Data) : Chaque processeur est complètement indépendant des autres et exécute les instructions sur des données différentes

# Architecture matérielle supportée

## ☛ Multi-processeurs :

- ☛ **SIMD** (Single Instruction Multiple Data) : tous les processeurs exécutent les mêmes instructions, mais sur des données différentes.
- ☛ **MIMD** (Multiple Instructions Multiple Data) : Chaque processeurs est complètement indépendant des autres et exécute les instructions sur des données différentes.
- ☛ **Pipeline** : les différentes unité d'exécution sont mises en chaîne et font chacune une partie du traitement.
- ☛ **Architectures fortement couplées** : architecture à mémoire commune
- ☛ **Architectures faiblement couplées** : chaque processeur possède sa propre mémoire locale.
- ☛ **Architectures mixtes** : différents niveaux de mémoire (commune et privée)

# Évolution

- ☛ **Système fermé** : Le système n'accepte pas des ajouts (ou ayant une extensibilité réduite)
  - Pour lui ajouter d'autres fonctionnalités, il faut remettre en cause sa conception et refaire une archive (système complet)
  - c'est l'exemple de UNIX, WINDOWS, etc.
- ☛ **Système ouvert** : Le accepte des ajouts de fonctionnalités et des abstractions sans avoir à repenser le système ou même sans avoir à l'arrêter (extension à chaud)
  - Pour cela, le système doit être basé sur une conception modulaire.

# Contrainte de temps réel

- ☛ Certains automatismes (particulièrement) nécessite des contraintes de réponse en temps réel.
- ☛ Toute réponse reçus au delà du temps requis ne sera plus valide
- ☛ Le SE doit alors gérer cette automatisme et répondre dans les délais.
- ☛ On parle alors de **systèmes à temps réel**.
- ☛ Les contrainte des temps sont de deux types :
  - **Contraintes strictes (dures)**
  - **Contraintes relatives (molles)**

# Chapitre 1 : Introduction aux Systèmes d'exploitation

**Dr. Mandicou BA**

mandicou.ba@esp.sn

<http://www.mandicouba.net>

Diplôme Universitaire de Technologie (DUT, 2<sup>e</sup> année)  
Diplôme Supérieure de Technologie (DST, 2<sup>e</sup> année)  
en Informatique



ECOLE SUPERIEURE POLYTECHNIQUE

[www.esp.sn](http://www.esp.sn)

