



# Systemes d'exploitation

## Partie 1: Introduction

Jalil BOUKHOBZA

ENSTA-Bretagne / Lab-STICC

Email : [jalil.boukhobza@ensta-bretagne.fr](mailto:jalil.boukhobza@ensta-bretagne.fr)

# PLAN DU COURS

- 1. Introduction aux systèmes d'exploitation**
- 2. Les processus**
3. Concurrency et exclusion mutuelle
4. Communication inter processus
5. Threads
6. Ordonnancement
7. Gestion de la mémoire
8. Systèmes de fichiers

Cours  
d'aujourd'hui

# Partie 1 : Introduction

1. Préambule
2. Qu'est-ce qu'un système d'exploitation ?
3. La double fonction d'un système d'exploitation
4. Décomposition fonctionnelle d'un système d'exploitation
5. Les différents types d'architecture de systèmes d'exploitation
6. Autre classification des OS

# 1. PREAMBULE

/ Ce cours a pour objet :

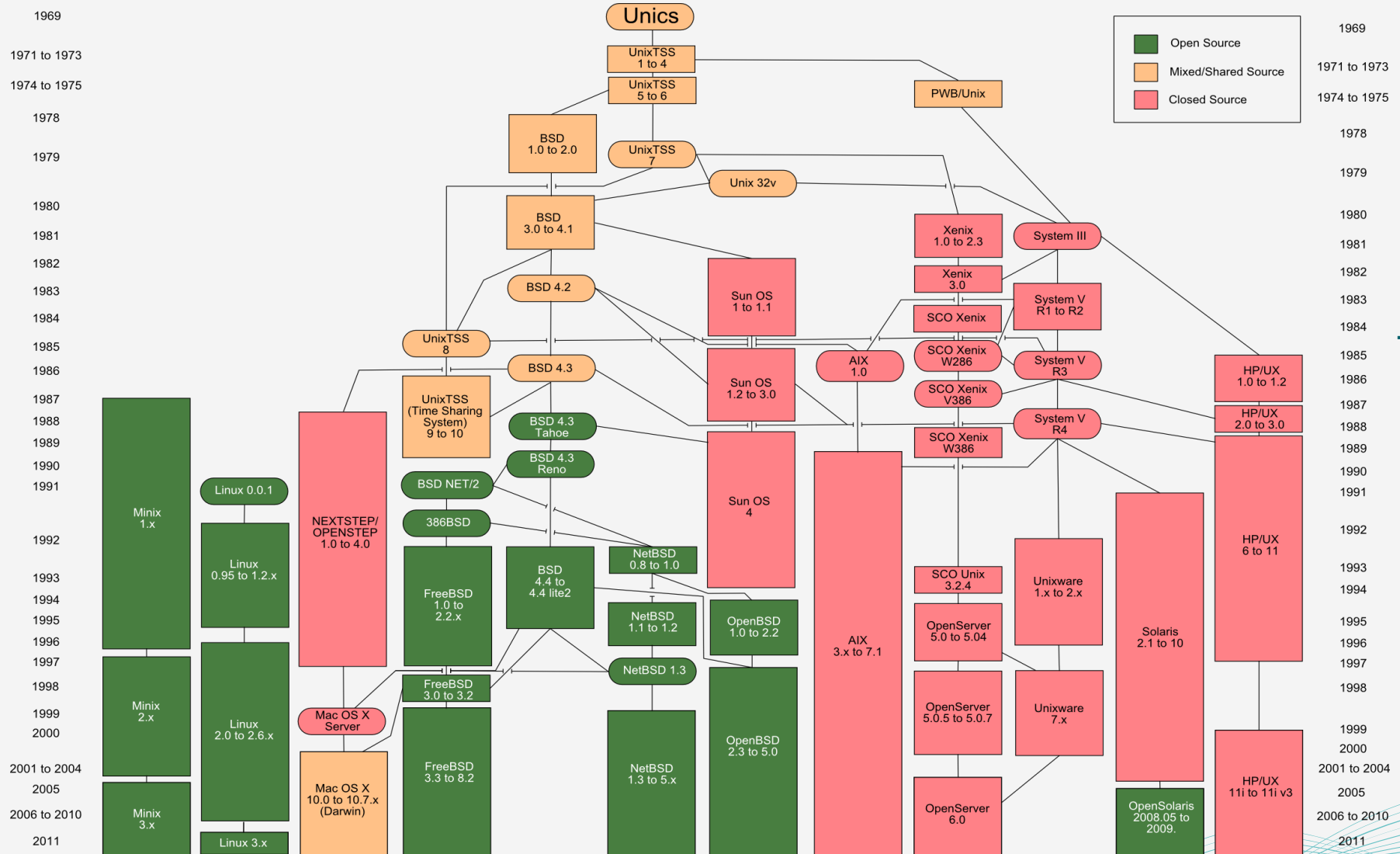
- / de décrire le rôle et le fonctionnement d'un système d'exploitation mono-processeur et multi-tâches
- / d'apprendre à manipuler certains concepts de base

● Ces concepts de base seront essentiellement illustrés au travers du système **UNIX/Linux** qui est un système :

- très homogène,
- très riche,
- très souple.

# Les systèmes Unixiens

## Unics (UNiplexed Information and Computing Service)



Source: Wikipedia / Levenez.com

## 2. QU'EST-CE QU'UN SYSTEME D'EXPLOITATION (1)

/ Un ordinateur est constitué :

1. Du **matériel**

/ Dispositifs physiques

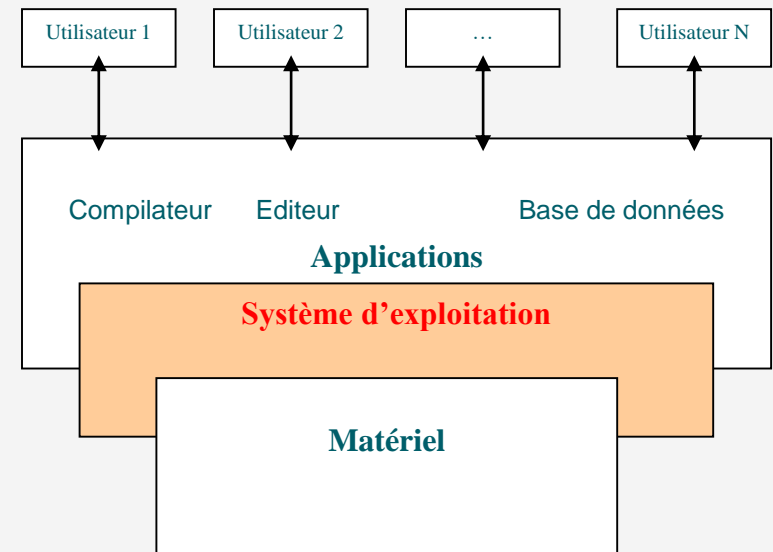
/ Langage machine

2. D'un **système d'exploitation**

3. De **programmes**

/ Programmes système

/ Programmes d'applications



## 2. QU'EST-CE QU'UN SYSTEME D'EXPLOITATION (2)

Le matériel (Hardware): Les dispositifs physiques constituent la couche la plus basse du matériel.

- / le processeurs,
- / la mémoire principale,
- / des disques,
- / des imprimantes
- / des interfaces de connexion réseau ,
- / ...

## 2. QU'EST-CE QU'UN SYSTEME D'EXPLOITATION (3)

Le système d'exploitation (Operating System ou OS ou SE): c'est le plus important des programmes systèmes :

1. Il contrôle les ressources de l'ordinateur.
2. Il libère le programmeur de la complexité du matériel.

Il se compose:

- / **D'un noyau (kernel)**: partie la plus critique d'un OS. Il permet aux éléments matériels et logiciels de communiquer entre eux, de fonctionner ensemble et de former un tout. Pour ces raisons, il est le premier logiciel chargé en mémoire.
- / **Des outils système**: partie permettant à l'utilisateur de tirer profit de l'OS, de gérer les périphériques, les configurer ...En bref, ils fournissent une interface d'accès au système.

Exemple de tâche:

- / LIRE UN BLOC DU FICHIER, ...



## 2. QU'EST-CE QU'UN SYSTEME D'EXPLOITATION (4)

Les programmes (Software, applications): ils sont écrits

- / par les utilisateurs
  - / ou par les éditeurs de logiciels
- / But: résoudre des problèmes spécifiques tels que:
- / le traitement des données commerciales,
  - / les calculs scientifiques,
  - / etc.

### 3. LA DOUBLE FONCTION D'UN SYSTÈME D'EXPLOITATION

/ Un système d'exploitation permet de répondre à deux besoins qui ne sont pas forcément liés :

1. le système d'exploitation en tant que machine étendue (ou « machine virtuelle »),
2. le système d'exploitation en tant que gestionnaire de ressources.

/ Exemple de systèmes d'exploitation:

/ Linux, Mac OS, Windows 9X, 10, XP, Seven, MS-DOS, MINIX, etc.

## 3.1 En tant que machine étendue

/ Le système d'exploitation correspond à « *l'interface* » entre les applications et le matériel.

/ De ce point de vue le système d'exploitation peut être assimilé à une machine étendue ou virtuelle **plus facile à programmer ou à utiliser** que le matériel :

/ Un programmeur va utiliser le système d'exploitation par l'intermédiaire “ d'appels système ”.

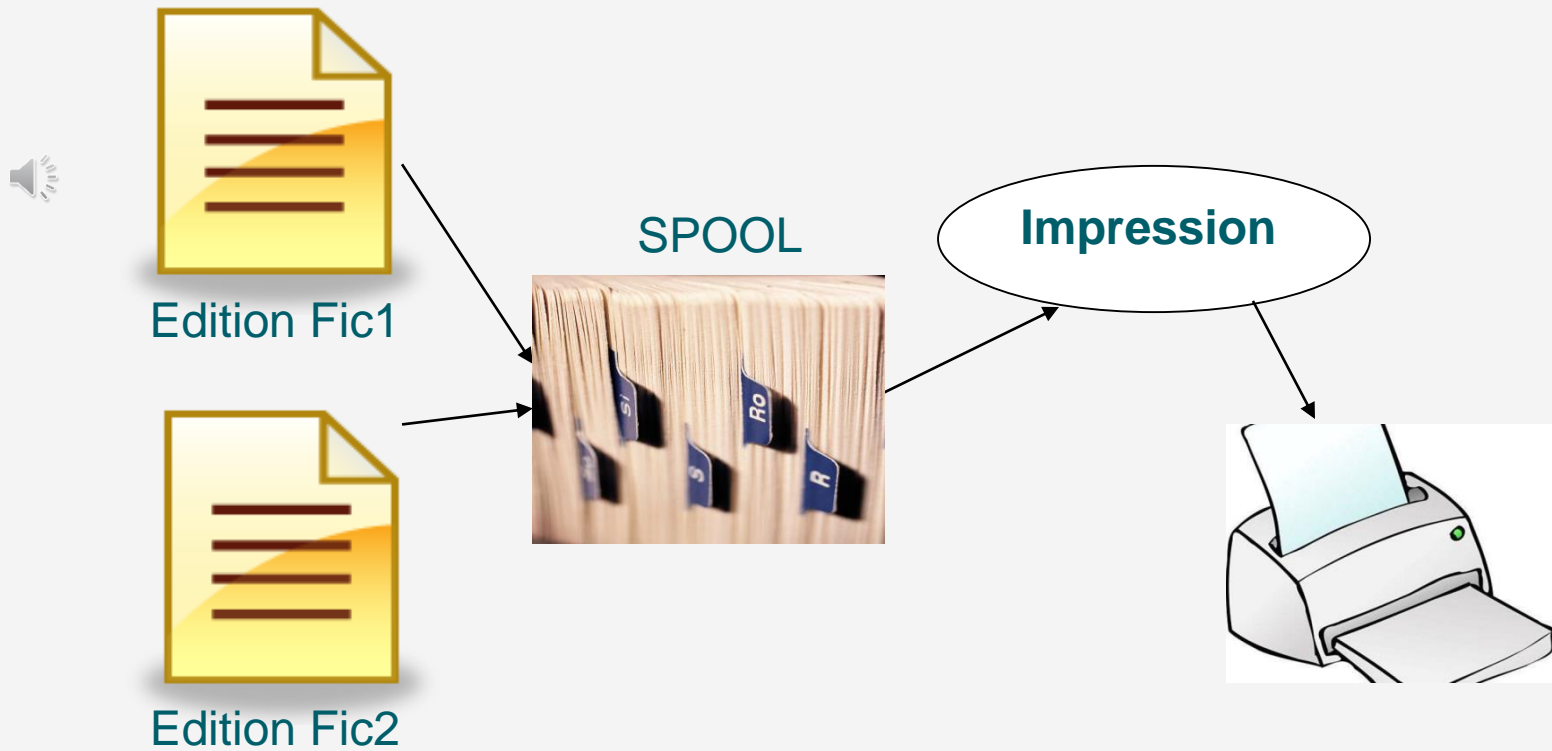
/ Un utilisateur peut lui aussi – dans une certaine mesure – manipuler un système d'exploitation, sans pour autant avoir à créer un programme (commandes shell).

## 3.2 En tant que gestionnaire de ressources

- / Les différents composants d'un ordinateur doivent coopérer et partager des ressources.
  - / Dans cette optique, le travail du système d'exploitation consiste à :
    - / ordonnancer,
    - / contrôler l'allocation des ressources :
      - / processeurs,
      - / mémoires,
      - / périphériques d'E/S,
      - / ...
- entre les différents programmes qui y font appel.

## 3.2 En tant que gestionnaire de ressources (2)

### Exemple : Gestion des impressions



## 3.2 En tant que gestionnaire de ressources (3)

- / Pour chacune des ressources d'un ordinateur, le système d'exploitation doit :
  - / connaître à tout moment l'utilisateur de la ressource,
  - / en accorder l'usage de manière équitable,
  - / éviter les conflits d'accès entre les différents programmes ou utilisateurs.
- / Les deux tâches essentiels du Système d'exploitation en tant que gestionnaire des ressources sont :
  - / Le **partage** des ressources.
  - / La **protection** de l'accès aux ressources.

## 4. LES FONCTIONS DE BASE D'UN SYSTÈME D'EXPLOITATION

1. **La gestion des processus**
  - / qui correspondent à l'exécution des programmes.
2. **La gestion de la mémoire**
  - / qui permet de gérer les transferts entre les mémoires principale et secondaire.
3. **Le système de fichiers**
  - / qui offre à l'utilisateur une vision homogène et structurée des données et des ressources : disques, périphériques.
4. **Les entrées-sorties**
  - / qui correspondent aux mécanismes qu'utilisent les processus pour communiquer avec l'extérieur.

## 4. Autres fonctions d'un système d'exploitation

### / Les réseaux d'ordinateurs

/ avec les protocoles de communication, d'interconnexion et d'application.

### / Les systèmes répartis

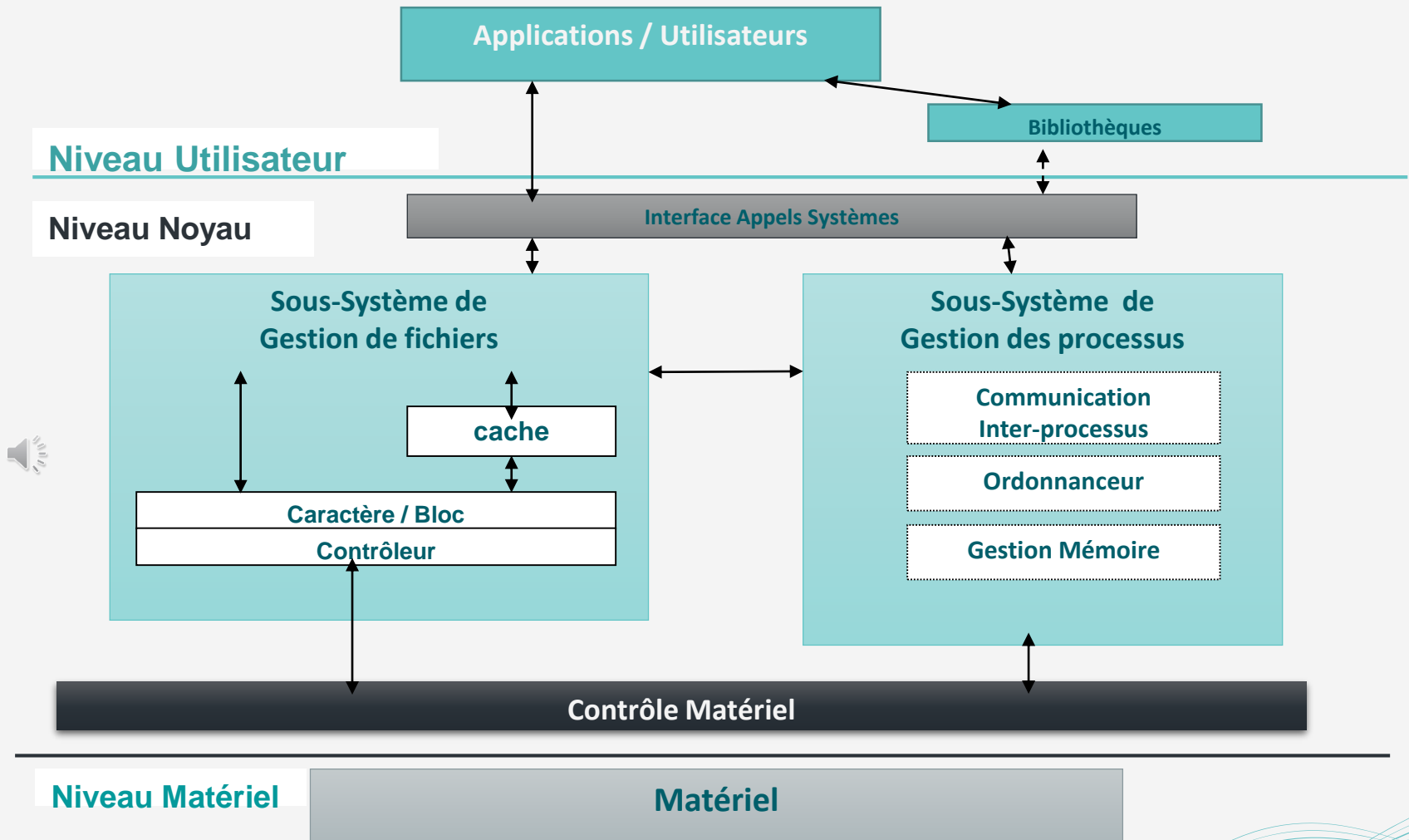
/ avec les protocoles d'appels de procédures à distance (RPC)

/ ou les objets distribués.

### / Les systèmes de fenêtrage graphiques.



# Une architecture du noyau



# 5. Les différents types d'architecture de systèmes d'exploitation

## / Différents types de systèmes d'exploitation:

- / Monolithiques

- / Modulaire / Multicouches

- / Micronoyaux

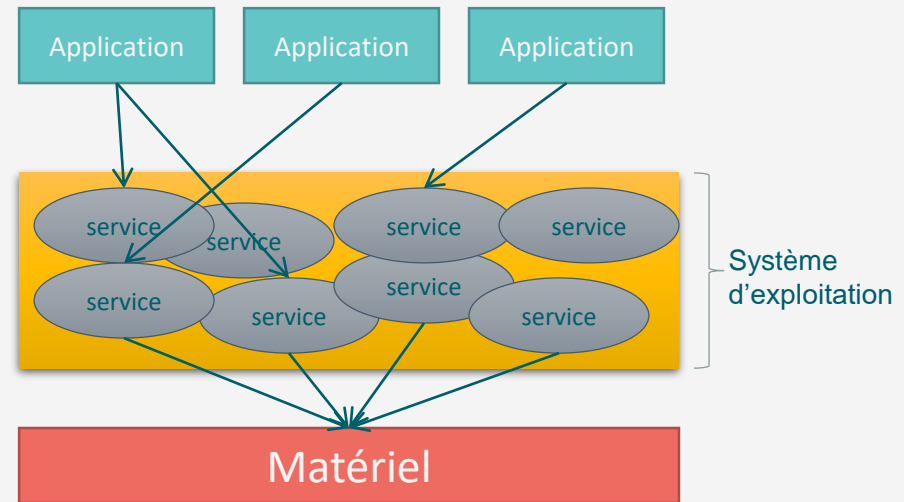
- / Exonoyaux

- / Machine virtuelle

/ Le noyau est généralement exécuté dans un espace mémoire séparé de l'espace des applications: **espace noyau**. Par opposition à **l'espace utilisateur**. Le passage entre ces deux espaces se fait via des appels systèmes. L'intérêt de cette séparation est que le système ne crash/se plante pas si une application plante.

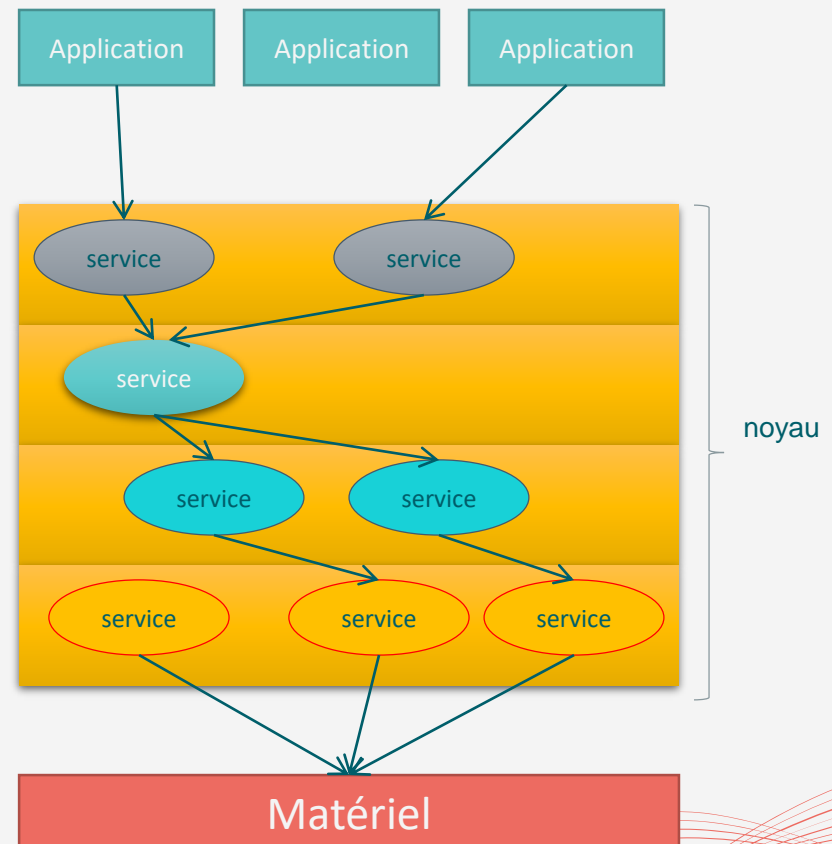
# OS Monolithiques

- / Un seul bloc contenant l'ensemble des services système (mode noyau).
- / Facilité de conception
- / Performance peut être au RDV
- ...
- / Code dur à maintenir
- / Usine à gaz !!
- / Dos, très vieux UNIX et Linux, etc.



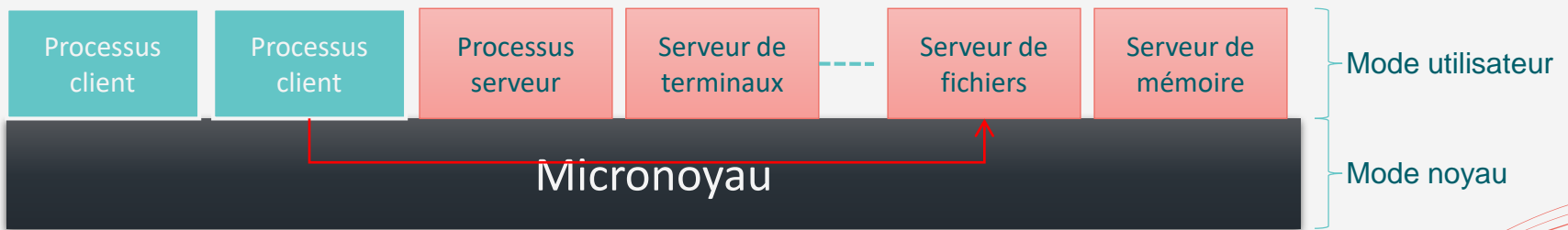
# OS Multicouches

- / OS organisé en **hiérarchie de couches**. Chacune construite sur la base des services offerts par la couche inférieure.
- / Facilité de conception et de développement
- / Code plus organisé et maintenable.
- / Chargement des fonctionnalités à la demande
- / Linux, BSD, SOLARIS



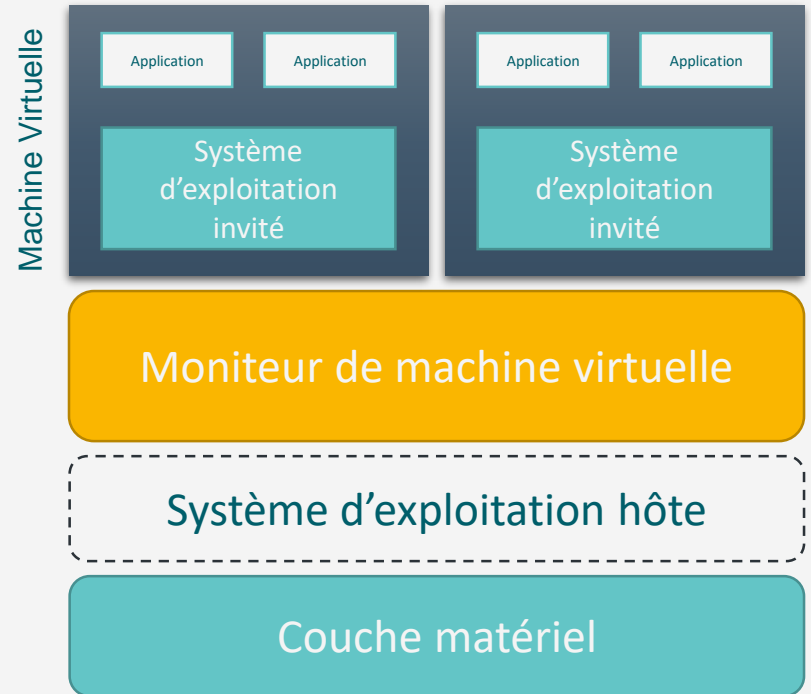
# OS Micronoyau

- / Déplace plusieurs fonctions de l'OS vers des « **processus serveur** » s'exécutant en mode utilisateur → réduction au maximum de la taille du code privilégié (en mode noyau).
- / **But**: gérer les communications entre applications et serveurs pour:
  - / **Renforcer la politique de sécurité**
  - / **Permettre l'exécution de fonctions système** (accès aux registres d'E/S, etc.).
- / Fiabilité augmentée: si un processus serveur « crash », le système continue à fonctionner et il est possible de relancer ce service sans redémarrer.
- / Modèle facilement étendu à des systèmes distribués
- / MAC OS X, GNU HURD, WindowsNT



# OS Machine virtuelle

- / Possibilité de mettre plusieurs OS sur une machine physique.
- / Le **moniteur de machine virtuelle** (hyperviseur) intercepte les instructions privilégiées envoyées par l'OS invité, les vérifie (politique de sécurité) et les exécute.
- / XEN, VMWare, VBox, etc.



## 5. Autre classification des OS

- / **OS temps partagé**: garantir le partage équitable du temps processeur et des ressources dans le but de maximiser le temps de traitement et de réduire le temps de réponse moyen.
- / **OS temps réel**: garantir les temps de réponse
  - / **Systèmes à contraintes souples/molles**: systèmes acceptant des variations minimales de temps de réponse (systèmes multimédias)
  - / **Systèmes à contraintes dures**: gestion stricte du temps pour conserver l'intégrité du système (déterminisme logique et temporel et fiabilité)
- / **OS embarqué**: OS prévus pour fonctionner sur des machines de **petite taille**, (PDA ou des appareils électroniques autonomes: sondes spatiales, robot, ordinateur de bord, etc.), possédant une **autonomie réduite** → gestion avancée de l'énergie + ressources limitées ... cours de