
SYS-S5 Kernel - Environnements

4/12

FLORIAN CLIQUET
NOTE TAKING OF ONLINE RESOURCES

9 juin 2024



Plagiarism Mention

We attest that the content of this document is original and stems from our personal reflections.

Sommaire

Introduction	4
1 Kernel - Environnements	5
1.1 Rappel	5
1.2 Noyau (Kernel)	6
1.2.1 Monolithique	6
1.2.2 Micro-Noyau	6
1.2.3 Monolithique vs Micro-Noyau	8
1.2.4 Hybride	9
1.3 Environnement Informatique	10
1.3.1 Personnel	10
1.3.2 A temps partagé	11
1.3.3 Client-serveur	11
1.3.4 Peer-to-Peer (P2P)	11
1.3.5 Distribué	12
1.3.6 Cloud computing	12
1.3.7 Systèmes en Grappes (Clustered)	12
1.3.8 Virtualisation	13
1.3.9 Conteneurisation	14

Introduction

This document provides an overview of SYS-S5 concepts written by Cliquet Florian. It's a set of notes on multiple online resources.

1 Kernel - Environnements

1.1 Rappel

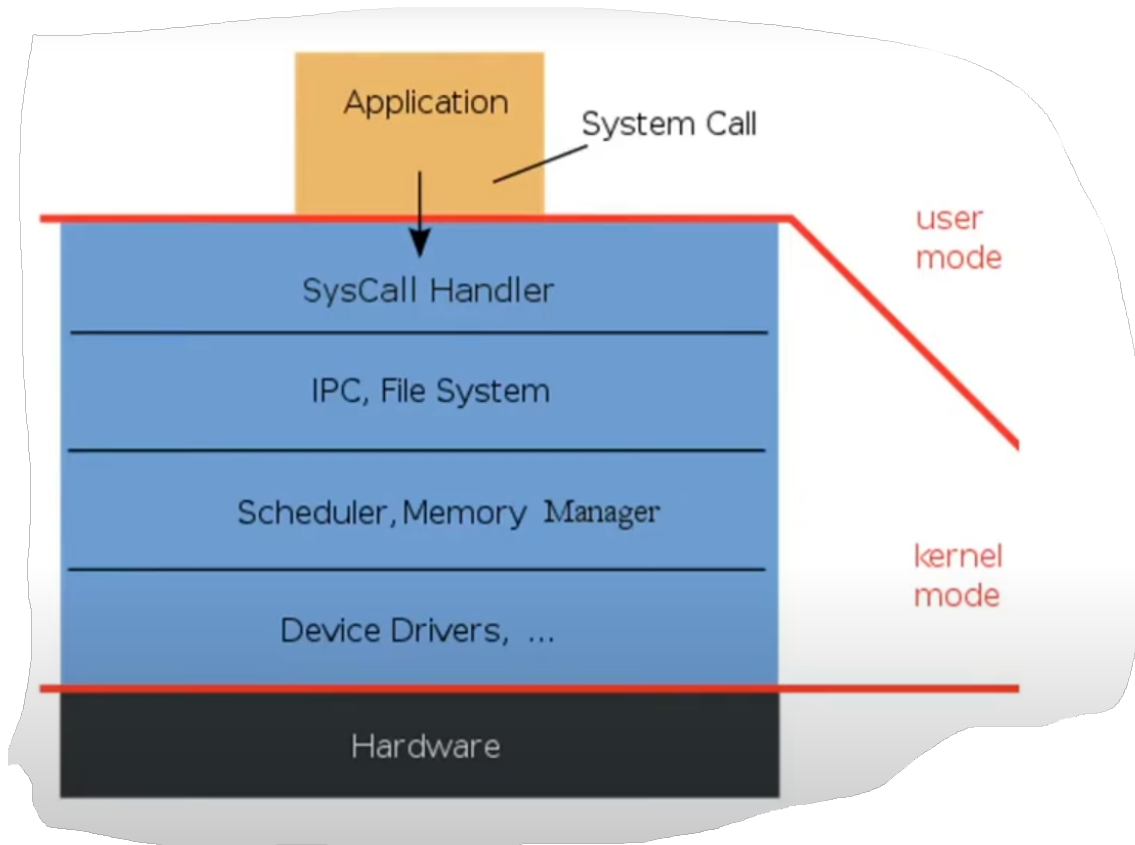


FIGURE 1 – 2 modes

Il existe deux modes, pour l'ouverture d'un fichier par exemple :

— **Mode Utilisateur :**

- Application
- System Call

— **Mode Noyau (Kernel) :**

- SysCall Handler
- File System
- Device Drivers
- Hardware
- Memory Manager
- SysCall Handler

1.2 Noyau (Kernel)

1.2.1 Monolithique

Ce type de noyau qui contient tous les services est appelé noyau **monolithique**.

Avantages :

Rapide : moins de temps à gaspiller en basculant d'un mode à l'autre (utilisateur/-noyau).

Dans l'exemple précédent, un seul passage au mode noyau et tout le traitement se fait dans ce mode.

Inconvénients :

Difficile à ajouter ou modifier des services.

Comme les services communiquent directement entre eux, chaque ajout ou modification d'un service peut influencer la communication entre lui et les autres services.

Exemple : UNIX

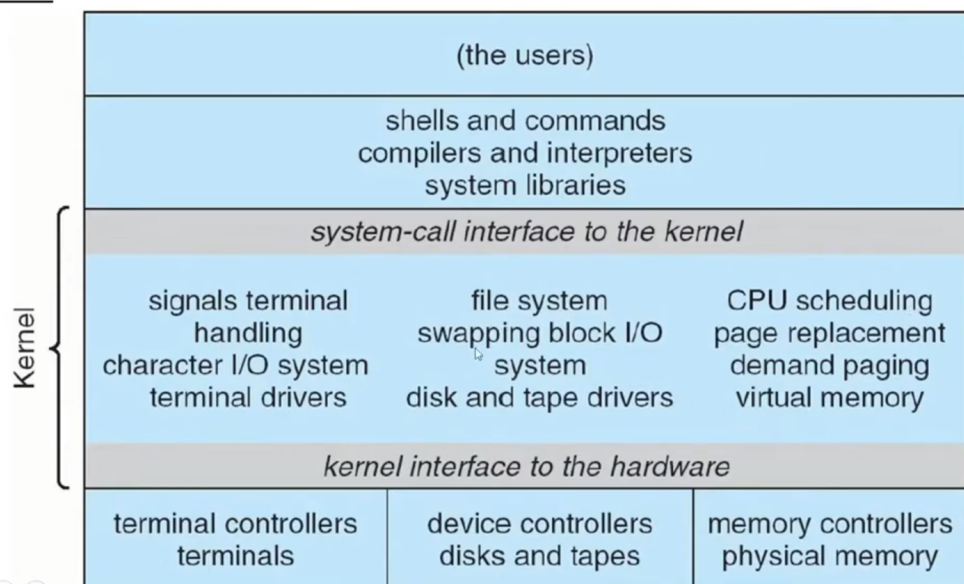


FIGURE 2 – Unix

1.2.2 Micro-Noyau

Le **micro-noyau** consiste à placer juste les services essentiels dans le noyau. Le reste est placé avec les applications utilisateurs.

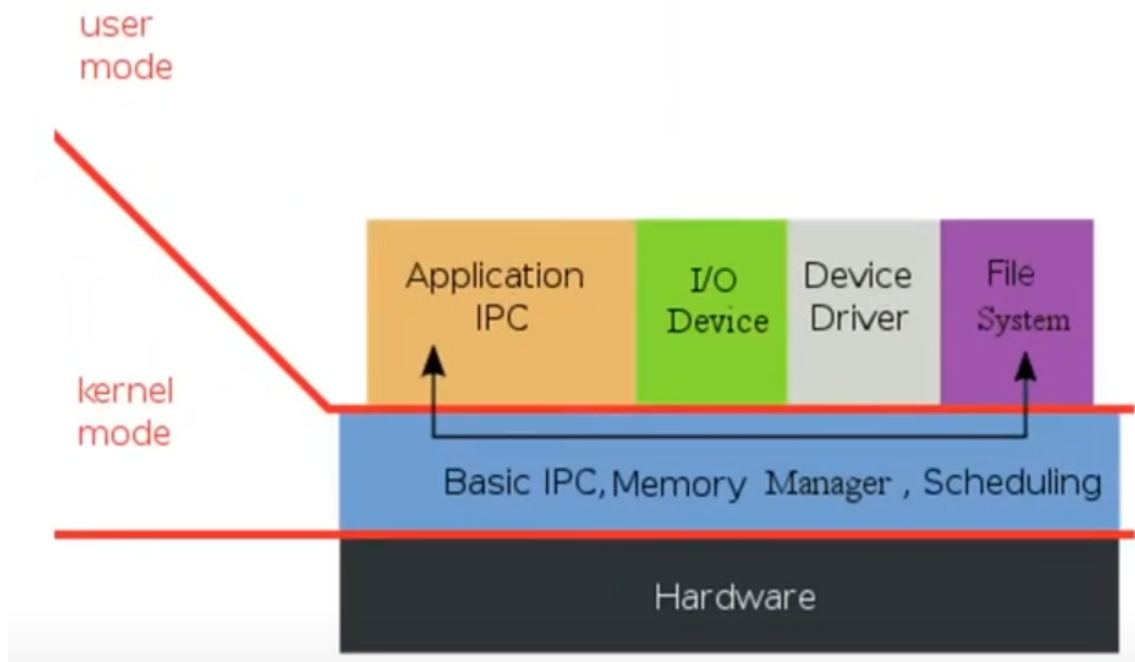


FIGURE 3 – micro-noyau

— **Avantages :**

Facilité d'ajouter ou modifier des services.

Comme les services ne communiquent pas directement entre eux, la communication passe par le noyau, l'ajout ou la modification d'un service n'influence pas la communication avec les autres services.

Portabilité facile vers de nouvelles architectures, plus sécurisé et plus fiable (moins de code s'exécute en mode noyau)

— **Inconvénients :**

Lenteur : beaucoup de temps à gaspiller à basculer d'un mode à l'autre (utilisateur / kernel)

La communication entre les modules utilisateurs se fait par passage de message Pour l'ouverture d'un fichier par exemple :

— **Mode utilisateur :**

Application

— **Mode Noyau (Kernel) :**

Basic IPC, Scheduling

— **Mode utilisateur :**

File system

— **Mode Noyau (Kernel) :**

Basic IPC, Scheduling

- **Mode utilisateur :**
Device Drivers
- **Mode Noyau (Kernel) :**
Basic IPC, Scheduling
Hardware
Memory Manager

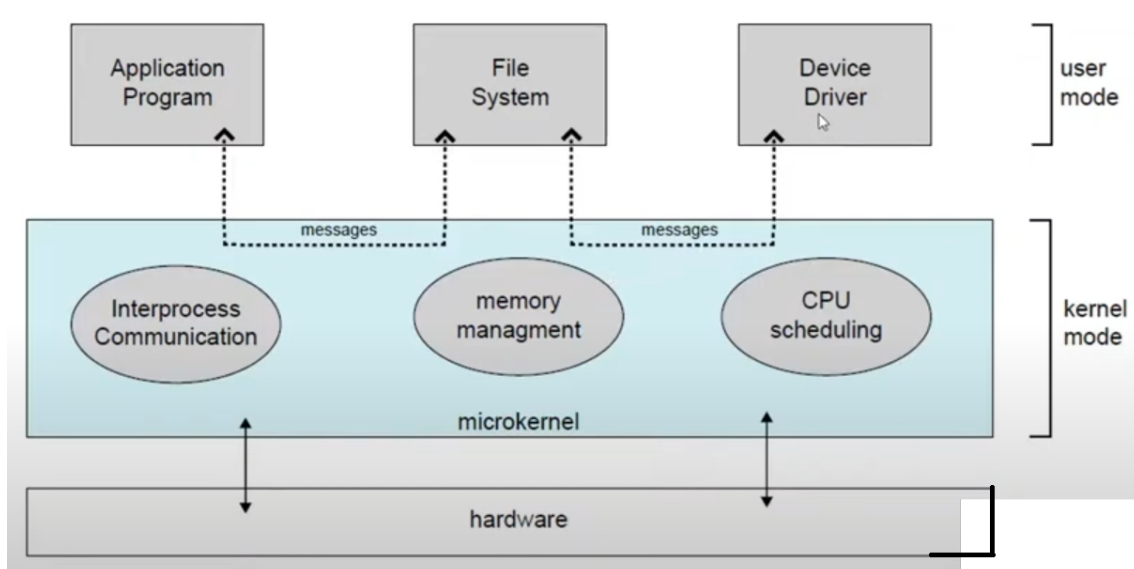


FIGURE 4 – Communication micro-noyau pour ouvrir un fichier

1.2.3 Monolithique vs Micro-Noyau

Monolithique vs Micro-noyau

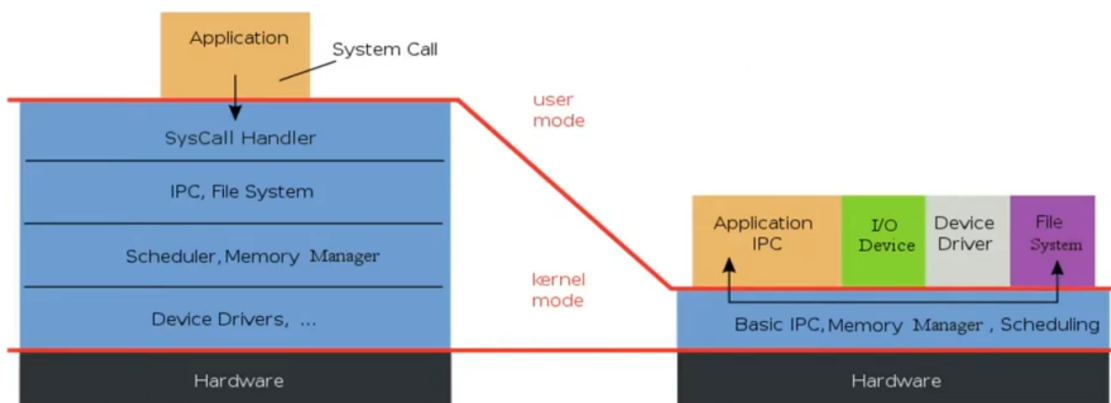


FIGURE 5 – Monolithique vs Micro-Noyau

Monolithique	Micro-Noyau
Un grand processus qui occupe un seul espace mémoire	Peut être divisé en plusieurs processus appelés serveurs
Moins de code à écrire pour le réaliser	Plus de code à écrire pour le réaliser
Il plante si un de ses services plante	Continue à fonctionner si un service plante
Communication via un seul fichier binaire statique	Communication via IPC

1.2.4 Hybride

Combine les aspects et les avantages du microkernel (Micro-Noyau) et du monolithic kernel.

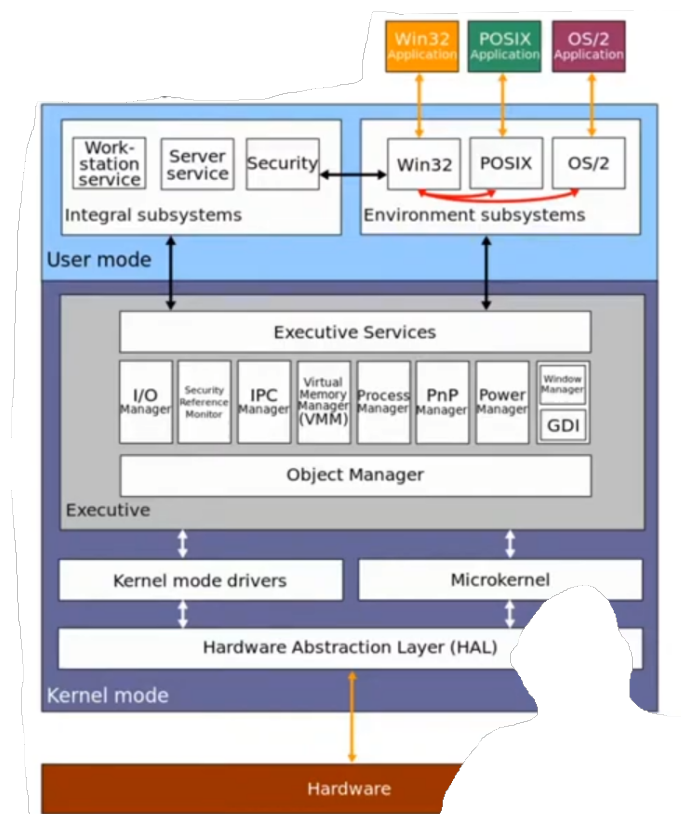


FIGURE 6 – Exemple : Microsoft NT kernel : Windows NT, 2000, XP, Server2008...

- Plus comme monolithique : services placés en mode kernel pour éviter le changement de modes
- Le microkernel est petit et ne contient que :
 - Premier-niveau de gestion des interruptions
 - Ordonnancement des threads
 - Primitive de synchronisation

1.3 Environnement Informatique

Un système informatique utilise différents appareils, disposés de différentes manières afin de résoudre différents problèmes.

Cela constitue un **environnement informatique** dans lequel des machines, avec différents matériels et systèmes d'exploitations, traitent et échangent des informations.

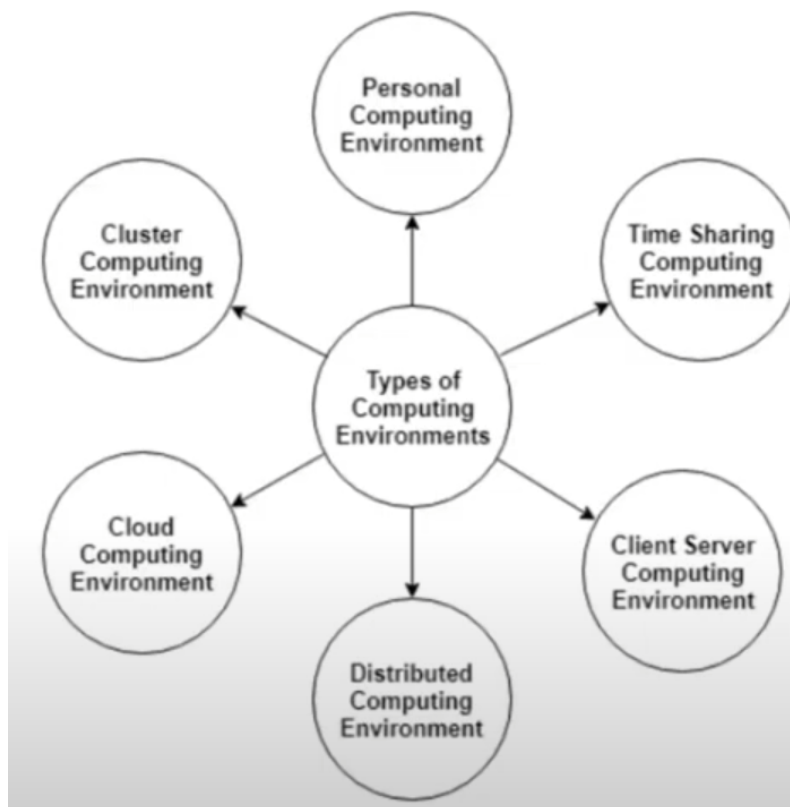


FIGURE 7 – Types of Computing Environments

1.3.1 Personnel

Dans l'environnement informatique personnel, il existe un seul système informatique. Tous les processus système sont disponibles sur l'ordinateur et y sont

exécutes.

Les différents appareils qui constituent un environnement informatique personnel sont les ordinateurs portables, mobiles, imprimantes, systèmes informatiques, scanners, etc.

1.3.2 A temps partagé

L'environnement informatique à temps partagé permet à plusieurs utilisateurs de partager le système simultanément.

Chaque utilisateur reçoit une tranche de temps et le processeur bascule rapidement entre les utilisateurs en fonction de celle-ci. Pour cette raison, chaque utilisateur pense être le seul à utiliser le système.

1.3.3 Client-serveur

Dans l'informatique client-serveur, le client demande une ressource et le serveur fournit cette ressource.

Un serveur peut servir plusieurs clients en même temps tandis qu'un client est en contact avec un seul serveur.

Le client et le serveur communiquent généralement via un réseau informatique, mais ils peuvent parfois résider dans le même système.

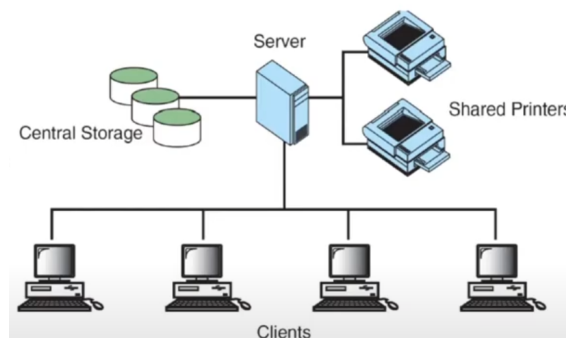


FIGURE 8 – Client-Serveur

1.3.4 Peer-to-Peer (P2P)

Dans l'informatique P2P, chaque noeuf, considéré comme un "peer", peut agir en tant que client, serveur ou les deux.

Le noeud peut diffuser une demande de service ou répondre aux demandes de service.

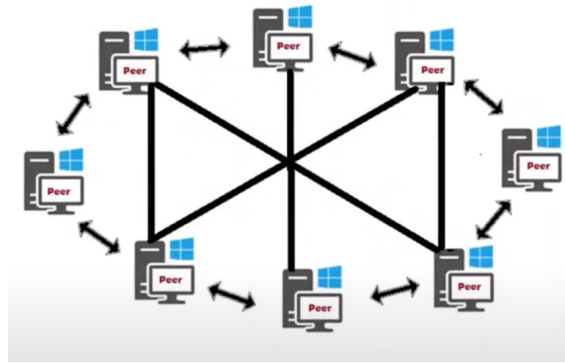


FIGURE 9 – Peer-to-peer schema

1.3.5 Distribué

Un environnement informatique distribué contient plusieurs noeuds physiquement séparés mais reliés entre eux à l'aide du réseau.

Tous les noeuds de ce système communiquent entre eux et participent comme un ensemble dans la gestion des processus.

Chacun de ces noeuds contient une petite partie du système d'exploitation distribué.

1.3.6 Cloud computing

Le Cloud computing fournit des serveurs de calcul, du stockage et même des applications en tant que service sur un réseau.

Les utilisateurs du cloud voient uniquement le service fourni et non les détails internes et la façon dont le service est fourni.

Le service peut-être :

- Software comme un Service (**SaaS**) une ou plusieurs applications disponibles via Internet
- Platform comme un Service (**PaaS**) pile logicielle prête à l'emploi via Internet
- Infrastructure comme un Service (**IaaS**) serveurs de calcul, ou stockage disponibles sur Internet

1.3.7 Systèmes en Grappes (Clustered)

L'environnement informatique en cluster est similaire à l'environnement informatique parallèle où plusieurs processeurs existent. Cependant, les systèmes en cluster sont créés par deux ou plusieurs systèmes informatiques individuels fusionnés pour qu'ils travaillent ensemble en parallèle.

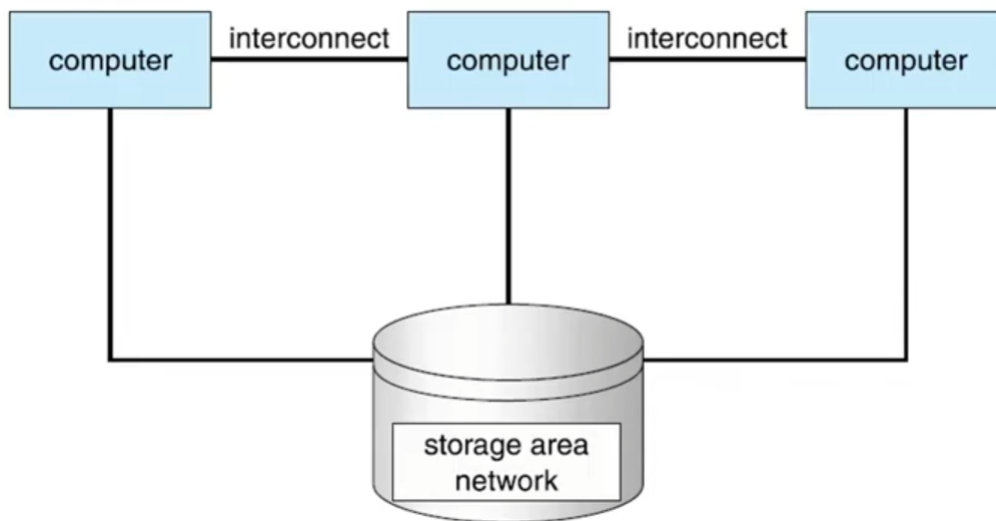


FIGURE 10 – Systèmes en Grappes

- Le partage du stockage se fait via **storage-area network (SAN)**
- La **haute-disponibilité**, qui survit aux pannes est assurée par :
 - **Grappes asymétriques** avec une machine en "hot-standby"
 - **Grappes symétriques** où les noeuds se surveillent mutuellement
- Les Grappes de calcul "**high performance computing**" (HPC) profite de :
 - Applications écrites pour utiliser la **parallélisation**
 - L'utilisation de **distributed lock manager (DLM)** pour éviter les conflits dans certaines opérations

1.3.8 Virtualisation

La virtualisation permet aux systèmes d'exploitation d'exécuter des applications dans d'autres systèmes d'exploitation.

- **Emulation** : utilisé lorsque le type de CPU source est différent du type de CPU cible (généralement plus lent)
- **Virtualisation** : SE compilé nativement pour le CPU exécutant les SE invités
- **Virtual Machine Manager** : fournit des services de virtualisation

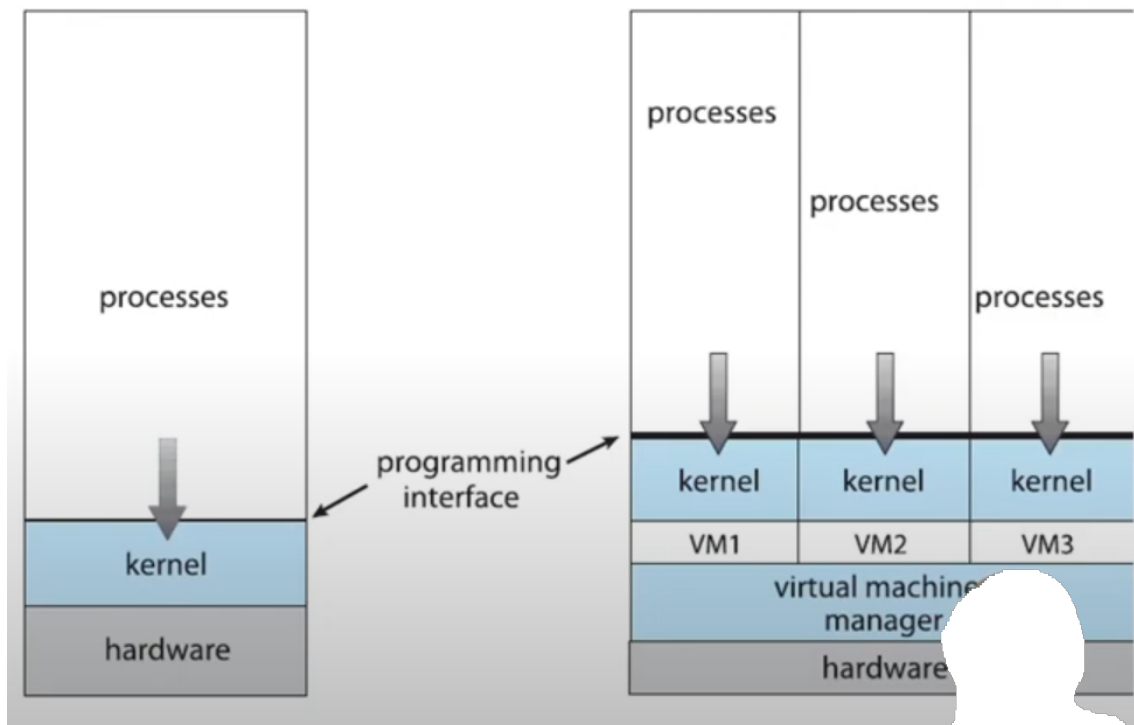


FIGURE 11 – Virtualisation

1.3.9 Conteneurisation

Docker est un logiciel libre permettant de lancer des applications dans des conteneurs logiciels.

Docker est un outil qui peut emballer une application et ses dépendances dans un conteneur isolé, qui pourra être exécuté sur n'importe quel serveur.

Cela permet d'accroître la flexibilité et la portabilité d'exécution d'une application.