



# Formation Virtualisation et containerisation



# Sommaire

- Prérequis Hardware
- Noyau
- VM vs Container
- VM
- CT
- Conclusion

# Prérequis Hardware :

## LE CPU

C'est l'unité logique et central par excellence au sein d'un pc

Il existe plusieurs philosophies d'architectures comme de microarchitectures.

Le CPU :

## ISA : Instruction Set Architecture

Nous compilons et travaillons pour une architecture précise

x86  
x86\_64  
ARM64  
PowerPC  
etc

Quel navigateur souhaitez-vous télécharger ? [?](#)

Firefox

Sélectionnez votre installeur préféré [?](#)

Windows 64-bit

- Windows 64-bit
- Windows 64-bit MSI
- Windows ARM64/AArch64
- Windows 32-bit
- Windows 32-bit MSI
- macOS
- Linux 64-bit
- Linux 32-bit

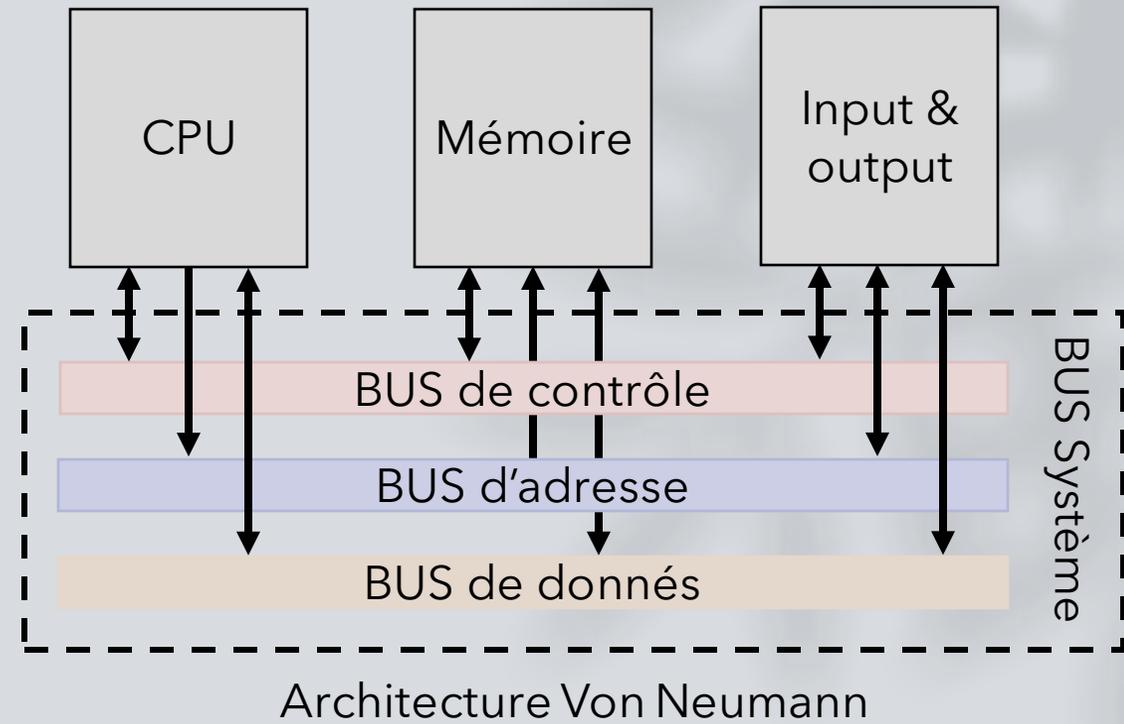
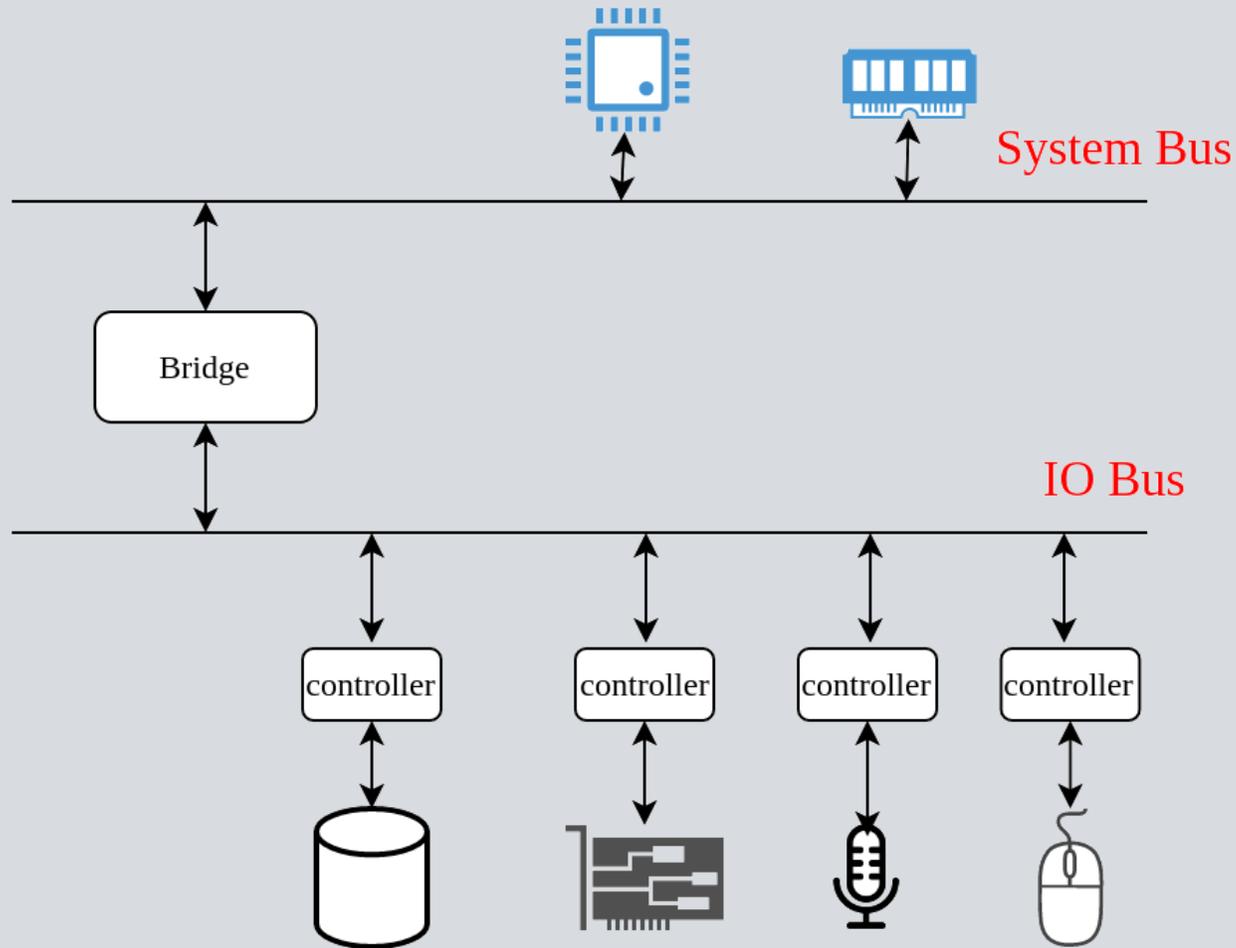
Jeux d'instruction :

Add x86  $\neq$  Add ARM32

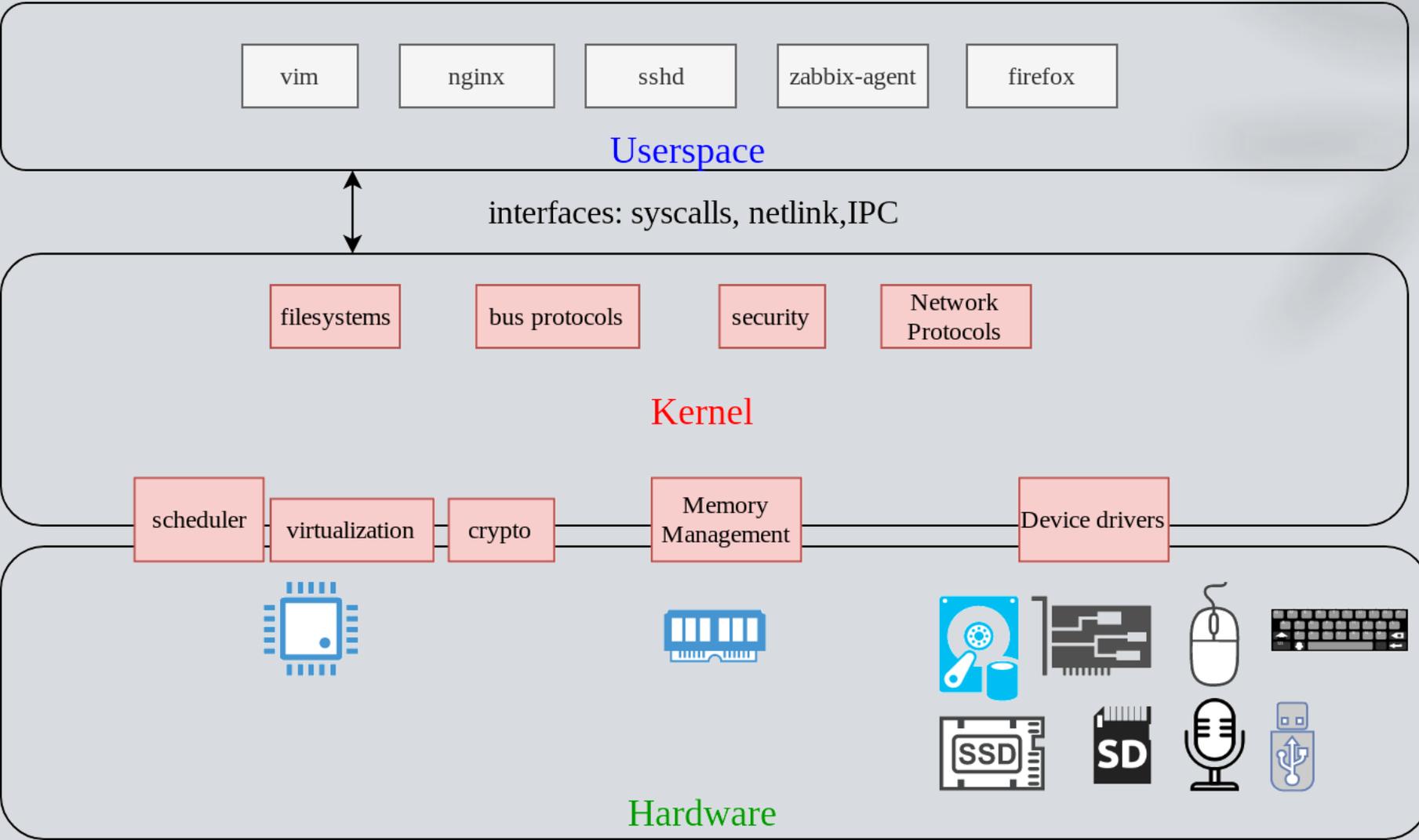
## **Micro-architecture**

Architecture propre à un constructeur qui ne modifie en rien les jeux d'instructions. C'est l'architecture matériel. (Ryzen, Alder Lake)

# L'organisation interne



# Noyau



GNU  
Windows

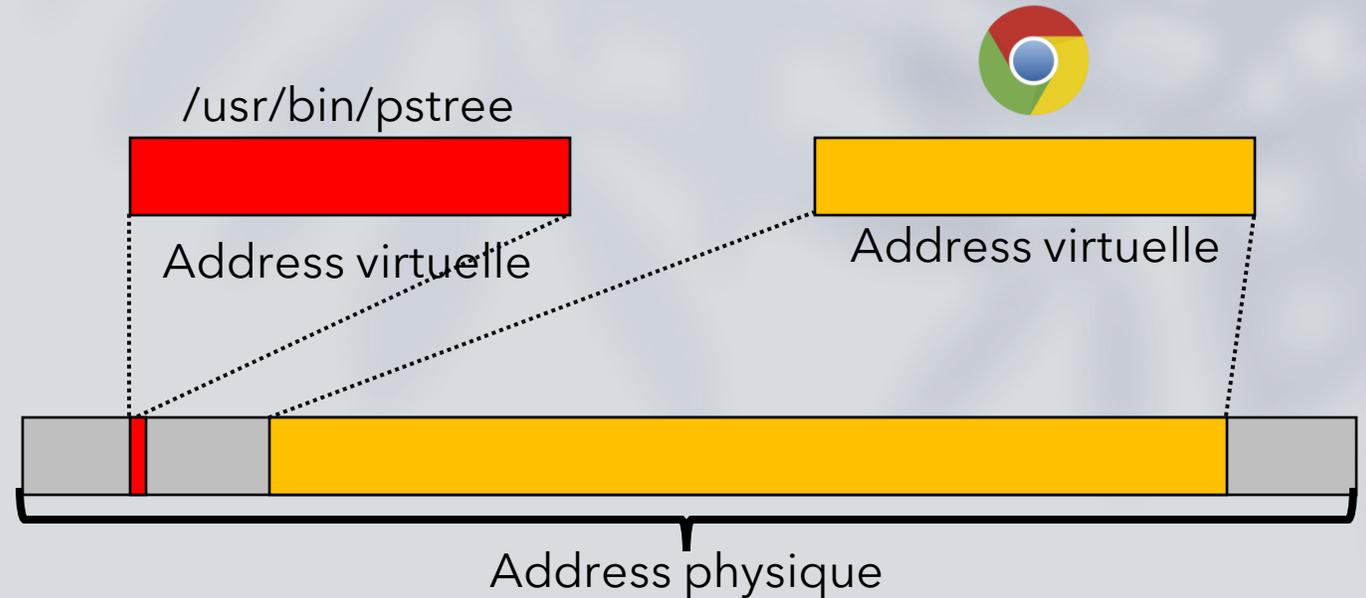
NT  
Linux

Hardware

# Un processus : quèsaco ?

```
> $ pstree
systemd—Xwayland—8*[{Xwayland}]
        |—alacritty—zsh
        |   |—15*[{alacritty}]
        |—alacritty—zsh—pstree
        |   |—15*[{alacritty}]
```

- Un espace virtuel isolé
- Un élément d'arbre (parents, enfants)



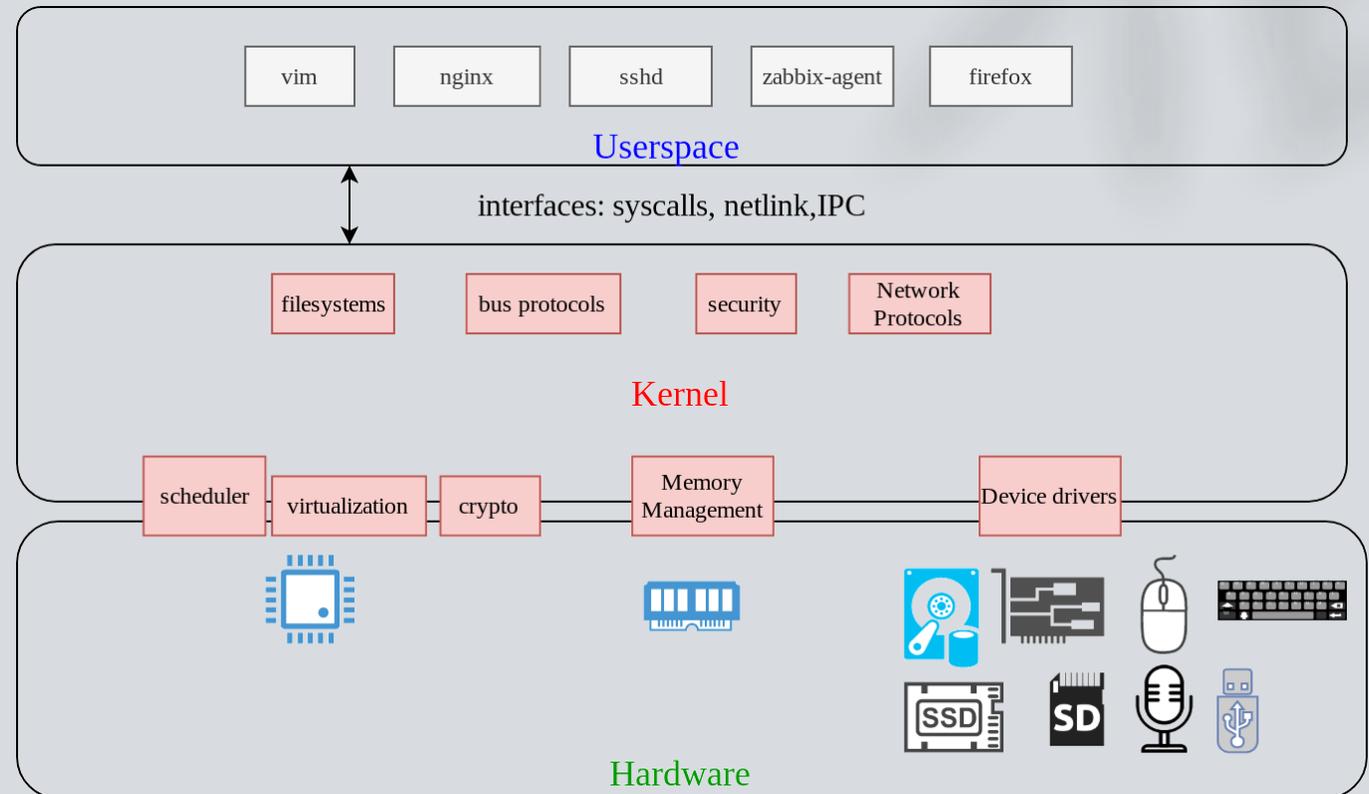
```
alacritty—zsh—test.sh—test.sh—test.sh—test.sh—test.sh—test.sh
```

# VM vs Container :

VM : reprend le fonctionnement à une machine standard à partir du noyau

CT : tourne sur le noyau de la machine, la séparation se fait au niveau du userspace

Pour rappel :



# VM

Emulation vs virtualisation

Fonctionnement

- La virtualisation

- La virtualisation du hardware

Possibilités et apports

Inconvénients

# Emulation vs virtualisation

Ce n'est pas magique le CPU a besoin de travailler pour exécuter le code de la VM !

Comment ? Tout dépend de la méthode

- **Emulation/Simulation :** traduction de blocs d'instructions vers d'autres blocs d'instructions
- **Full-virtualization :** Virtualisation assistée par matériel

## L'Emulation

Inspecte chaque instruction qui doit être exécuté et la traduit **avec une grosse détérioration des performances.**

Le système émulé ne tourne pas directement sur le hardware.

En pratique : Sert à émuler une architecture CPU différente de l'hôte.

Exemple :

Une Xbox360/PS3 (PowerPc)  $\neq$  Ton Ordinateur (x86\_64 || ARM64)

Add sur PowerPC  $\neq$  Add sur x86

## Full-virtualisation

Le code de la VM est exécuté directement via le processeur grâce à des instructions de virtualisation

Instructions de virtualisation diffèrent selon :

- architectures processeurs (ex : ARM64, x86, x86\_64)
- les constructeurs (ex : constructeurs x86\_64 : AMD et Intel) (AMD-V et Intel VT-X)

```
root@LAPTOP-06749VPD:~# lscpu
Architecture:                x86_64
CPU op-mode(s):              32-bit, 64-bit
Byte Order:                  Little Endian
Address sizes:               48 bits physical, 48 bits virtual
CPU(s):                      16
On-line CPU(s) list:         0-15
Thread(s) per core:          2
Core(s) per socket:          8
Socket(s):                   1
Vendor ID:                   AuthenticAMD
CPU family:                  25
Model:                       80
Model name:                  AMD Ryzen 9 5900HX with Radeon Graphics
Stepping:                    0
CPU MHz:                     3293.802
```

## Virtualisation avec assistance matériel

Exécution des instructions du processeur virtuel sur le processeur physique.

### Comment ça fonctionne ?

Le processeur de la machine physique possède un jeu d'instructions dédié (ex : `vmenter`, `vmexit`) qui permet de faire un changement de contexte entre le code de la VM à exécuté et le code de l'hôte.

```
root@cas:~# lscpu
Architecture:          x86_64
CPU op-mode(s):       32-bit, 64-bit
Byte Order:           Little Endian
Address sizes:        40 bits physical, 48 bits virtual
CPU(s):                4
On-line CPU(s) list: 0-3
Thread(s) per core:   1
Core(s) per socket:   4
Socket(s):             1
NUMA node(s):         1
Vendor ID:             AuthenticAMD
CPU family:            15
Model:                 6
Model name:            Common KVM processor
Stepping:              1
CPU MHz:               2094.772
```

# L' hyperviseur :

Une couche logicielle qui fait le lien entre les ressources physiques et virtuelles.  
C'est lui qui gère la virtualisation

- Il existe deux types d'hyperviseur :
  - **Type 1** : Tourne au dessus du matériel ( ex : *KVM, HYPER-V*)
  - **Type 2** : Tourne au dessus d'un OS ( ex : *QEMU*)



TP

<https://formation-container.minet.net>

# CONCLUSION

