Systems 3

Virtualization

Marcel Waldvogel

(Handout)

Department of Computer and Information Science
University of Konstanz

Winter 2019/2020

Photo by Kristopher Allison on Unsplash

Chapter Goals

- What is Virtualization used for?
- How does it work?
- What approaches are used?
- What are the differences?
- What are the hardware requirements?

Motivation

- Logical separation (modularization, independent migration)
- Security (sandboxing)
- Multiple environments
- Debugging
- System engineering
- OS development

Idea

Current abstractions (virtualizations)

- CPU → processes
- RAM → virtual memory
- Devices → common interface

Virtualization abstraction

Goal Exact copy of the system

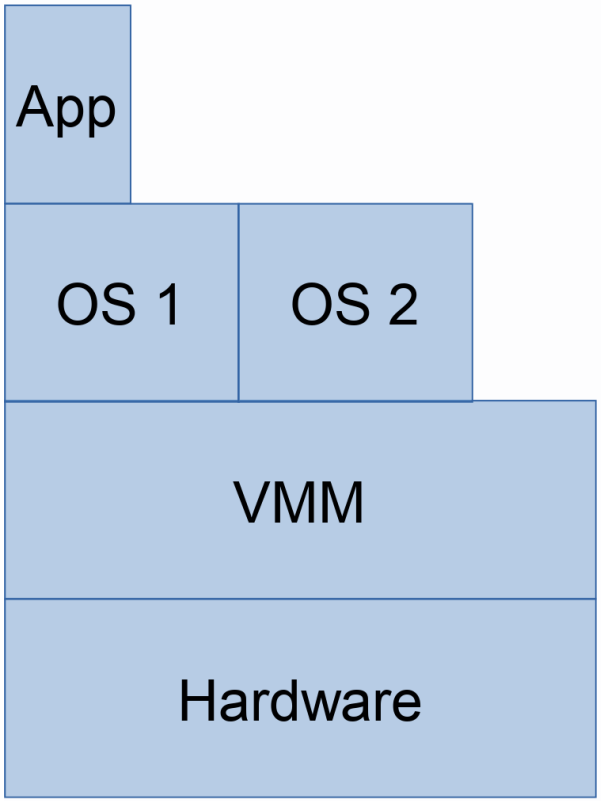
Means Encapsulate the entire OS in a 'meta-level OS'
(Hypervisor aka Virtual Machine Monitor (VMM))

Abstraction and Isolation

Mechanism	Goal
Functions	Group variables and code
Object files	Larger groups, separate compilation
Process	+ isolate memory and execution
Accounts	+ isolate files, other resources
`chroot`(2), `jail`	+ separate file system
Containers	+ isolate system (processes, libraries, ...)
Virtualization	+ separate OS, believes to run on hardware
Emulation	+ different (instruction set) architecture
Machine	+ real hardware

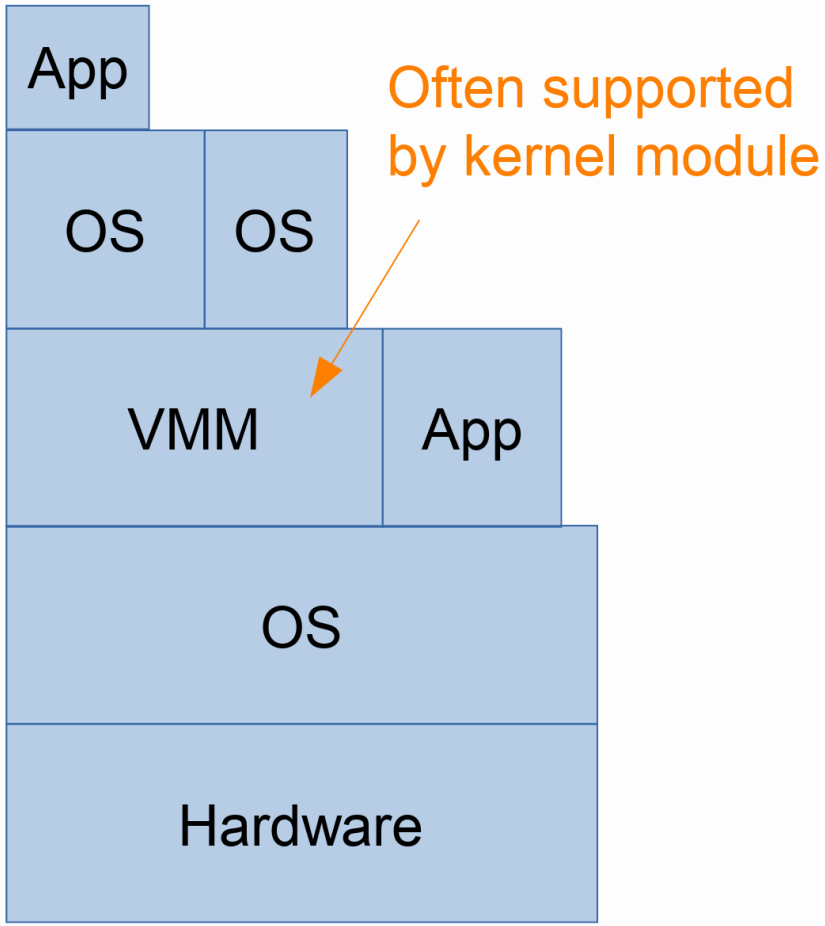
Positioning

Type 1 VMM



z.B.: Xen, Hyper-V, ESX

Type 2 VMM



z.B.: KVM, Virtualbox

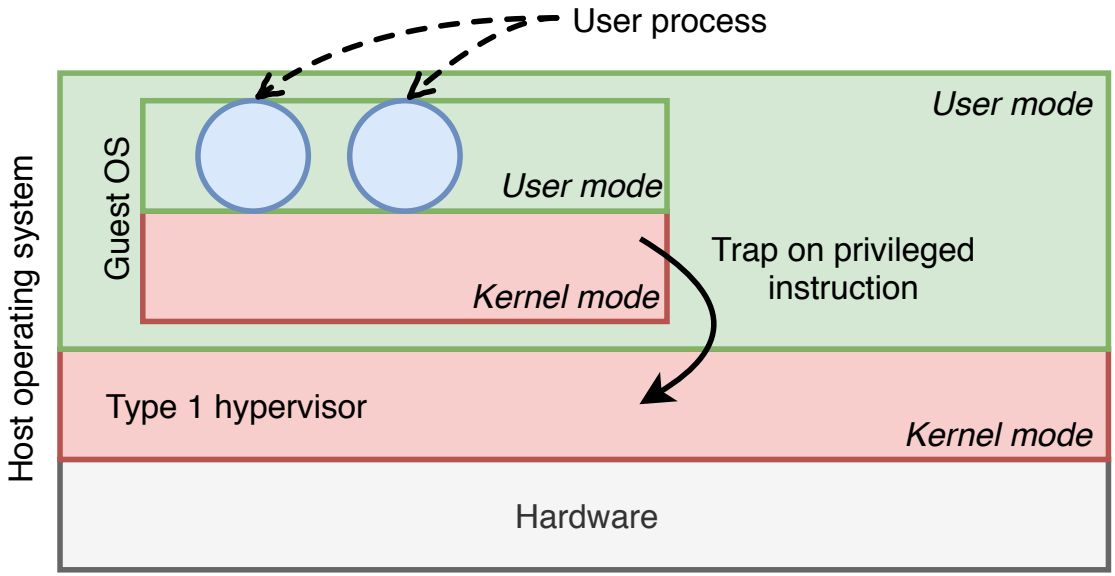
CPU virtualization

Idea

Run guest OS in user mode, trap and emulate privileged operations.

Implementation

VMM receives trap, securely emulates the effects of the instruction or memory access, and resumes execution. (x86 only since VT-x/AMD-V.)



Performance

Trapping expensive; therefore:

- Binary translation (in parts)
- Paravirtualization: Guest OS avoids problematic instructions/operations and cooperates with VMM.
- Direct hardware access (under control of IOMMU etc.).