Practical course: Advanced System Programming

Hypervisors

https://dse.in.tum.de/

Masanori Misono

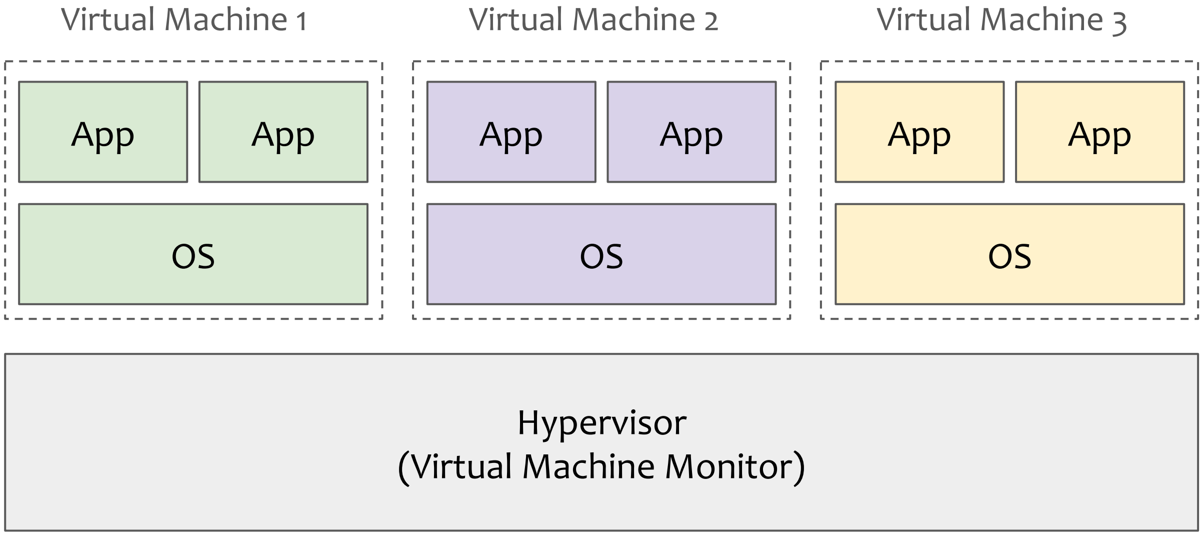
About this lab

- What you will learn
  - **Hardware-assisted virtualization**
    - Basics of Intel VT-x
  - **Linux KVM** and its ecosystem to implement a hypervisor on Linux

- What you will *not* learn
  - OS-based virtualization, aka container (docker, etc.)
  - Non-hw-assisted virtualization techniques (e.g., binary translation)

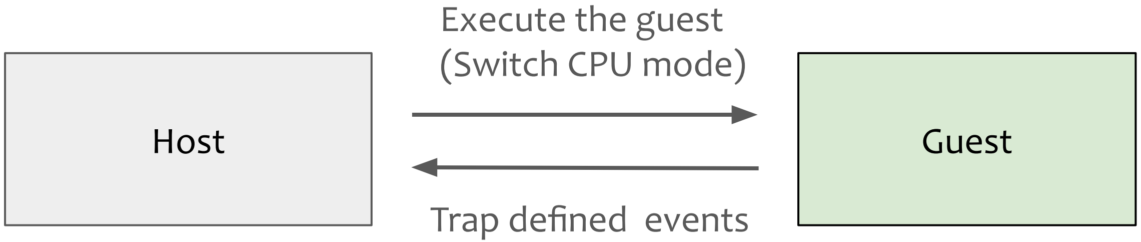
- This lab targets Linux / x86-64 environment

Virtualization



How to Virtualize

- Nowadays, most CPUs have **hardware-assisted virtualization** features
  - **Intel VT-x**, AMD-v, ARM VHE, RISC-V H extension…
- Main features
  - Introduce a new CPU mode for virtualization
    - A VM (guest) runs in the own address space, isolated from the host
  - Trap selective events in the guest, transfer control to the host
    - Interrupts, I/O instructions, …

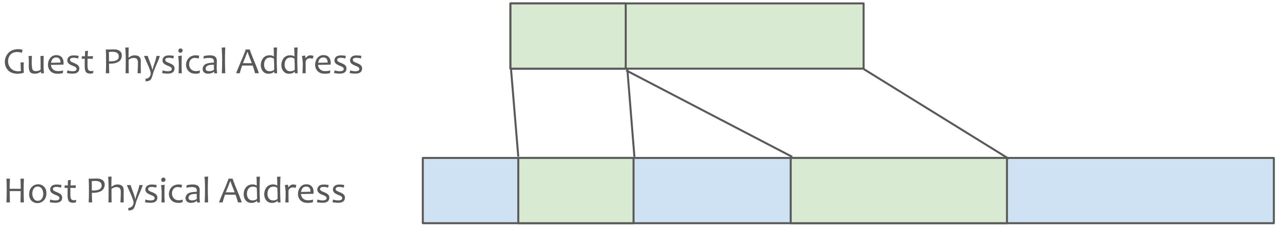


Execute the guest (Switch CPU mode)

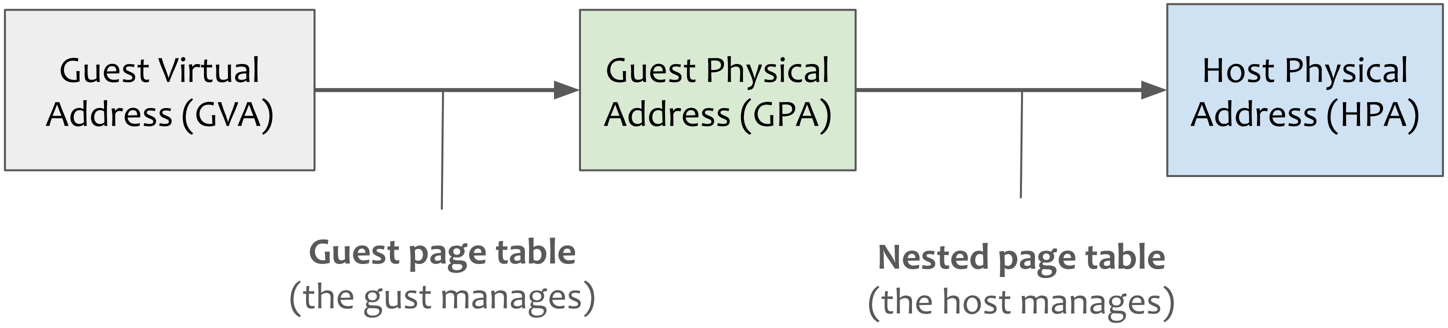
Trap defined events

Memory Virtualization with Nested Paging

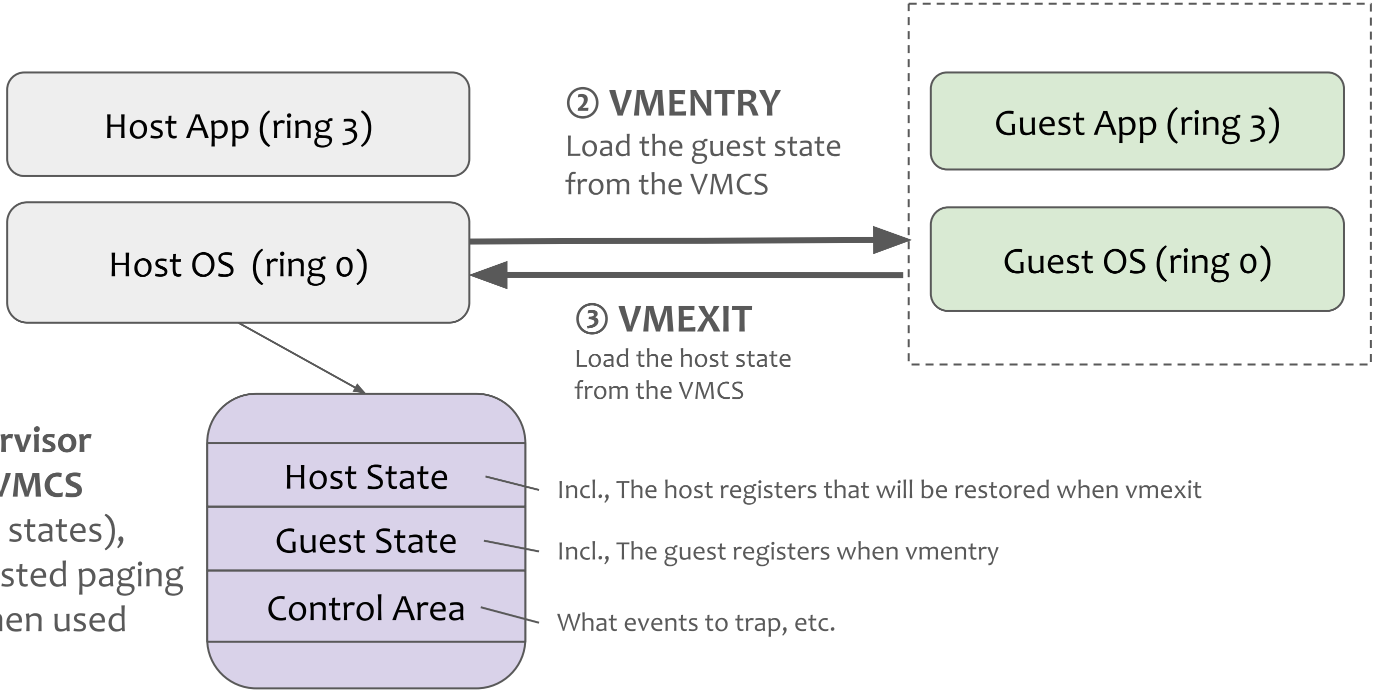
- The host needs to manage guest physical address



- Nested paging performs 2-level address translation for the guest


Guest Virtual Address (GVA)
Guest Physical Address (GPA)
Host Physical Address (HPA)
Guest page table
(the gust manages)
Nested page table
(the host manages)

Intel VT-x



Host App (ring 3)

Host OS (ring 0)

② **VMENTRY**
Load the guest state
from the VMCS

③ **VMEXIT**
Load the host state
from the VMCS

Guest App (ring 3)

Guest OS (ring 0)

① **The hypervisor configures VMCS** (VM control states), and **EPT** (nested paging for VT-x) when used

Host State — Incl., The host registers that will be restored when vmexit

Guest State — Incl., The guest registers when vmentry

Control Area — What events to trap, etc.

VMX Root mode

VMX non-Root mode

Overview

- ~~Virtualization 101~~
- Linux KVM (Kernel-based Virtual Machine)

Linux KVM (Kernel-based Virtual Machine)

- Make Linux as a hypervisor with hardware-assisted virtualization
  - Utilize existing Linux’s mechanism as much as possible (scheduling, etc.)
  - Provide generic API to userspace to implement hypervisor
    - KVM alone does not work as a stand-alone hypervisor!
- KVM-based hypervisors
  - QEMU/KVM[†], Firecracker, Crossvm, Cloud-hypervisor, …

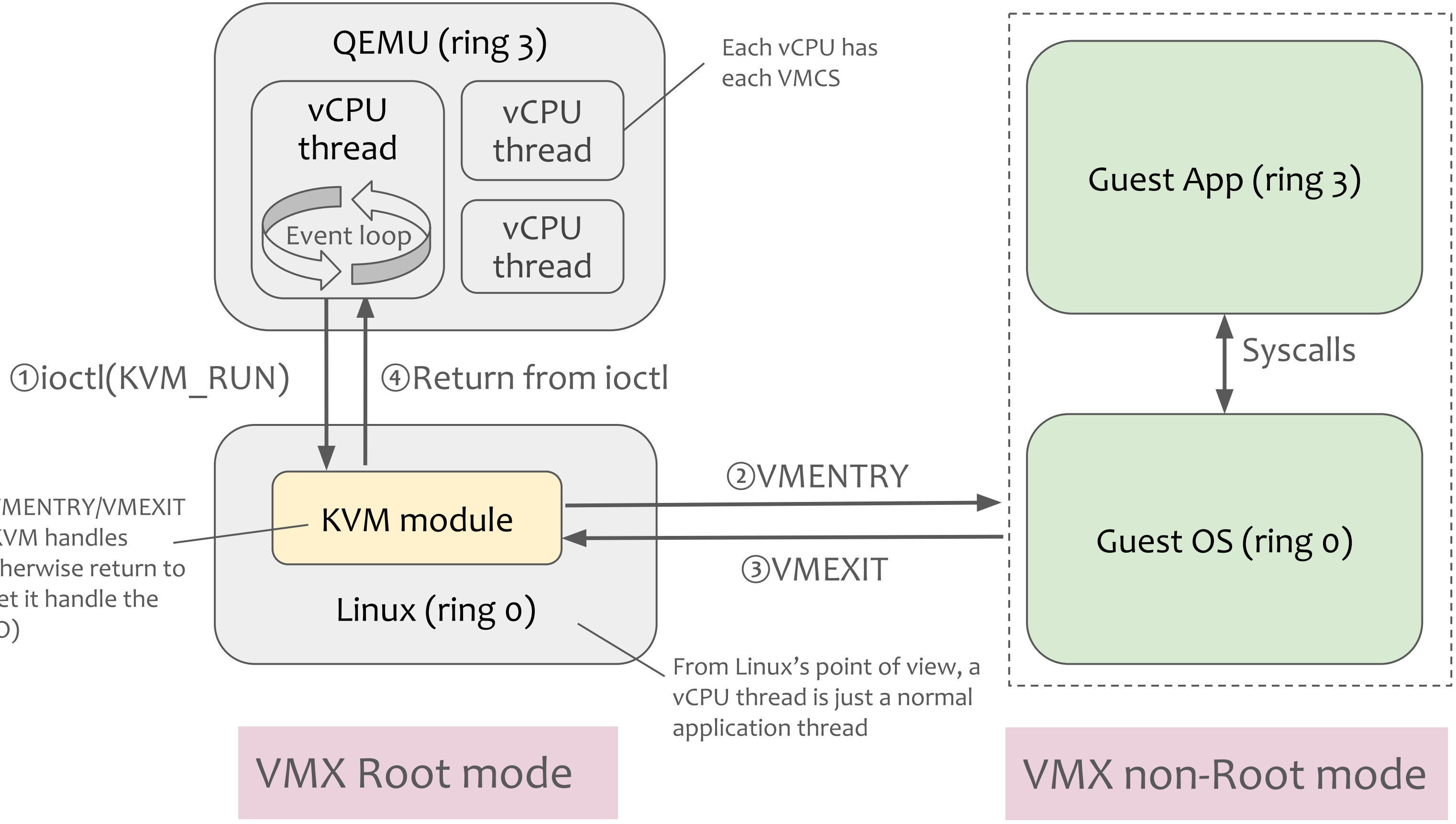
[†] QEMU can also work without KVM as a full system emulator

Quick Glance of KVM API

- KVM_CREATE_VCPU
  - Create vCPU
- KVM_SET_REGS
  - Configure the guest register states
- KVM_SET_USER_MEMORY_REGION
  - Configure the guest memory region
- KVM_RUN
  - Run a VM with the configured state
- …

KVM internally configures VMCS and EPT

QEMU/KVM workflow



Each vCPU has each VMCS

KVM manages VMENTRY/VMEXIT
Upon VMEXIT, KVM handles some events, otherwise return to the QEMU and let it handle the event (mainly I/O)

From Linux's point of view, a vCPU thread is just a normal application thread

Summary

- Hardware-assisted virtualization
  - Core component of modern virtualization
  - Example: Intel VT-x
- Linux KVM
  - Provides API to utilize hardware-assisted virtualization features in Linux
  - Many hypervisors use KVM nowadays

References

- Virtualization
  - Andrew Tanenbaum, Herbert Bos, “Modern Operating Systems” 5th Edition Chapter 7 Virtualization and the Cloud , Pearson Education, 2023.
  - Edouard Bugnion, Jason Nieh, Dan Tsafrir, “Hardware and Software Support for Virtualization”, Synthesis Lectures on Computer Architecture, 2017.
  - Gerald J. Popek , Robert P. Goldberg, “Formal Requirements for Virtualizable Third Generation Architectures”, Communications of the ACM Vol 17, No.7, 1979.
- KVM
  - Avi Kivity, Yaniv Kamay, Dor Laor, Uri Lublin and Anthony Liguori, “KVM: the Linux Virtual Machine Monitor”, Ottawa Linux Symposium 2007, 2007, https://www.kernel.org/doc/ols/2007/ols2007v1-pages-225-230.pdf
  - The Definitive KVM (Kernel-based Virtual Machine) API Documentation, https://docs.kernel.org/virt/kvm/api.html
- Intel VT-x
  - Intel® 64 and IA-32 Architectures Software Developer Manuals Volume 3 System Programming Guide, https://www.intel.com/content/www/us/en/developer/articles/technical/intel-sdm.html