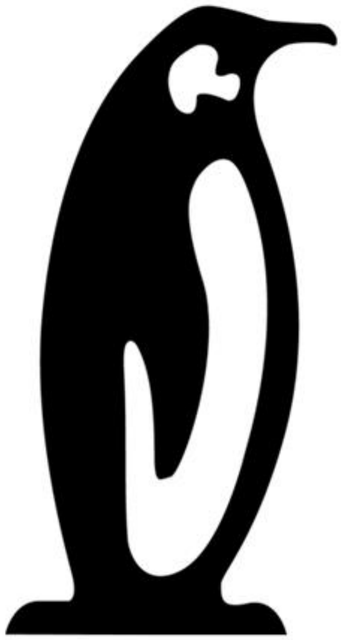


Linux Plumbers Conference

Vienna, Austria | September 18-20, 2024

OpenHCL: A Linux based paravisor for Confidential VMs

Chris Oo – Microsoft

LINUX
PLUMBERS
CONFERENCE Vienna, Austria / Sept. 18-20, 2024

Microsoft

What is a paravisor?

- Firmware component that runs inside the guest at a higher privilege level
- Provide emulation for unenlightened guests
  - APIC emulation and interrupt virtualization
- Provide services for guests
  - vTPM
  - Legacy emulated devices like serial
  - Device translation such as NVME to paravirt SCSI

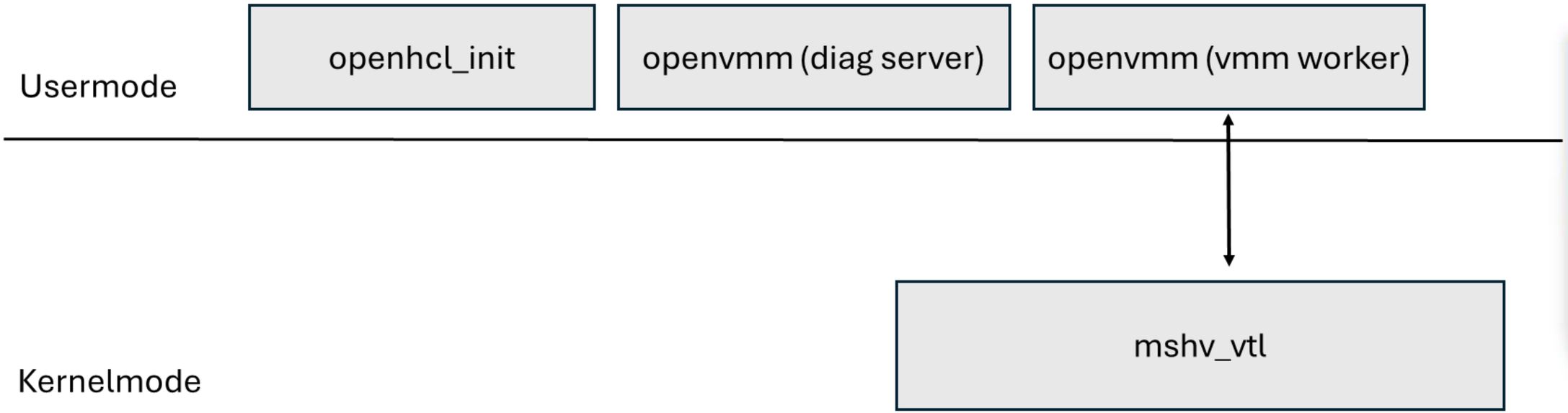
Why have a paravisor and not a svsm?

- Run guests that are not fully enlightened such as Windows and older Linux
- Provide device translation via standard device interfaces

OpenHCL overview

- Linux and usermode Rust based paravisor
- Open source later this year
- More details in BoF talk later today

OpenHCL architecture



Design philosophy

- Track upstream kernel
  - Aim to upstream all kernel patches or have a path to upstream
- Do as much in usermode as possible
  - Host the VMM itself in usermode
  - Device drivers in usermode
- Do as much in safe idiomatic rust as possible
- Rust async-focused usermode VMM
- Keep VMM code OS agnostic
  - Allows for running outside of OpenHCL

VMM worker

- Main process that acts as a VMM for the guest
- Handle exits from the platform
- Per vCPU executor hosting async tasks
- Interacts with mshv_vtl driver to perform VMM functions
  - Modifying register state
  - Accessing ram via mmap

Open discussion

- Could we have a single code base for both a SVSM and paravisor?
- Could we run the openvmm usermode in a SVSM?
- What learnings are there for OpenHCL that apply to a SVSM?
- How do TDISP devices interact with a guest?

Thanks!