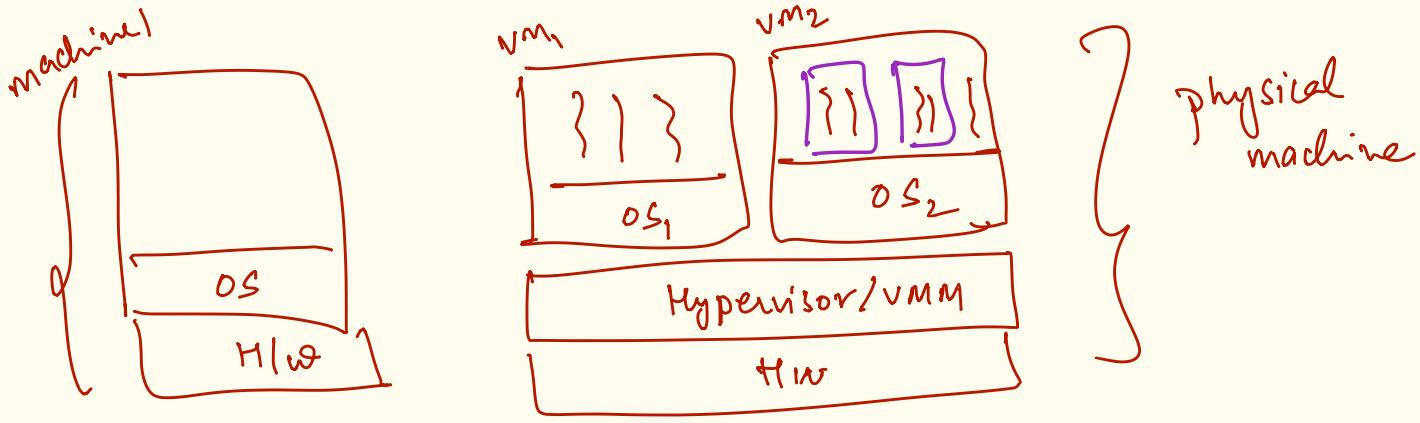
CS744 Lecture 27

Lab Quiz 2 today! 6th Nov. 5 pm.

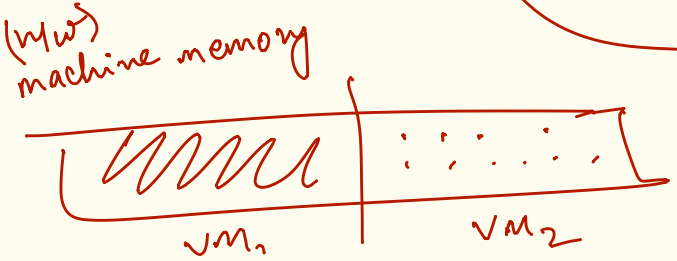
(*) VMs and beyond.

Q1: Given a machine which entity is the owner/ arbitrator of all resources? — operating system (OS).

Q2. With VMs, which entity is owner of all resources? phy. of the machine



e.g: lcr3 ← instruction to store memory value in CR3 regs. ~ pointer to the pgd. used by the mmu h/w for page table walks.



value is CR3 needs to be checked/ certified / translated.

(*) guest update → check → actual update.

mechanism to invoke this check (VMM fn).

x86

Ring 3 user lcr3 — ✗ trap interrupt

Ring 0 OS lcr3 — ✓

w/ VMs.

Ring 3 Apps — lcr3 ✗ trap

1 OS — lcr3 ✗ trap

0 VMM

interrupt handler

do
- checks on intended mem. address
- go or no go
  - actually update mCR3

- inst.
- state (mem. address) for update.

① trap-and-emulate (cpu virtualization for VMs)

- OSes have no idea of a VM-based execution
- update ~~critical~~ sensitive resource configs. via critical/sensitive instructions.
- execute OS at lower PL
- trap instructions
  - emulate intended behaviour in VMM trap handler.

invariant : critical inst^{n}. + lower PL

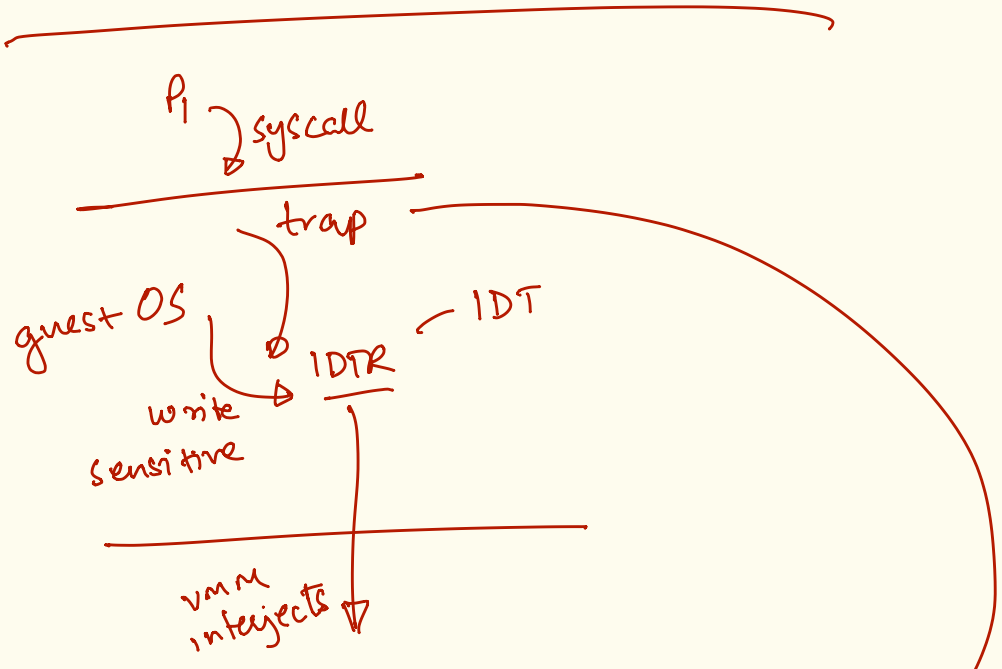
$\Rightarrow$ trap.

this is no true for all ISAs.

instructions

eg: x86 popf — used to set/reset interrupt flag.

does a noop at PL=1



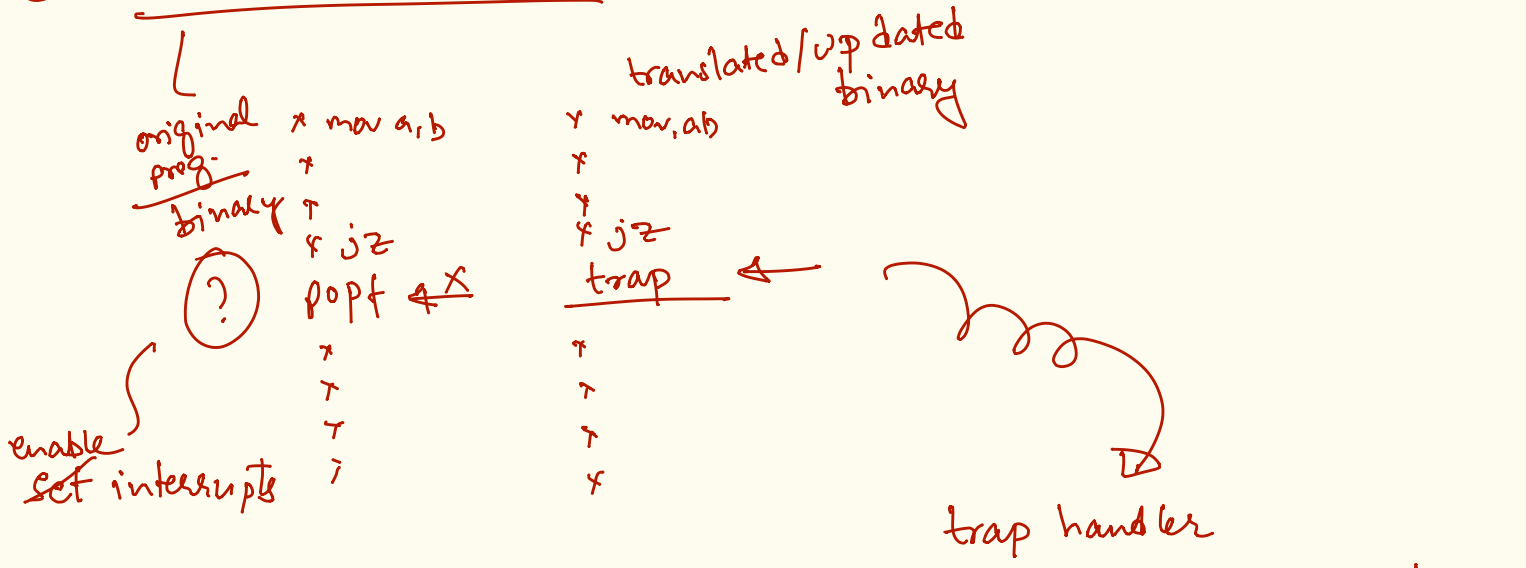
(*) guest OS$_1$: IDTR$_1$ loads IDTR w/ IDTR$_1$

or loads IDTR hyp.

interrupt handler of hypervisor

- check interrupt
- dispatch / call IDT f^{n}. of guest OS.

② binary translation / scan & patch.



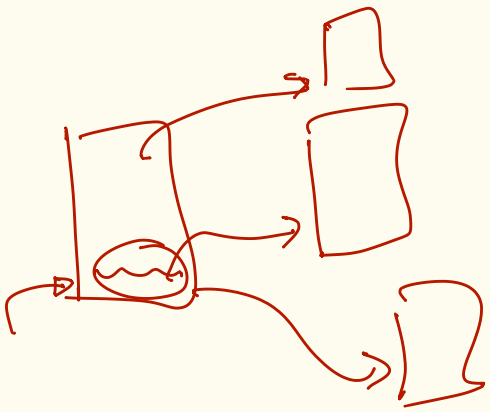
③ para-virtualization

- guest OS knows about the VM-env.
  - hypervisor.

adv:

(i) guest OS can skip / not use problematic instr. ≠ by design.
or embed scan-and-patch explicitly.

(ii) use ~~pass~~ an interface to pass information to the hypervisor via hypercalls.



update_pgtab(, all updates)

x86: 2 bits of the CS regs.
└ CPL — current priv. level

④ hardware-assisted virtualization of CPUs. (Intel VT-X / AMD VMX)

Ring 3 Apps.

Ring 0 guestOS —enter root VMM→ VMM

non-~~vmx~~ root VMX ←exit root VMM— root VMX

→ VMCS ← 4KB state config. register buffer.

→ stores state of guest OS for restore

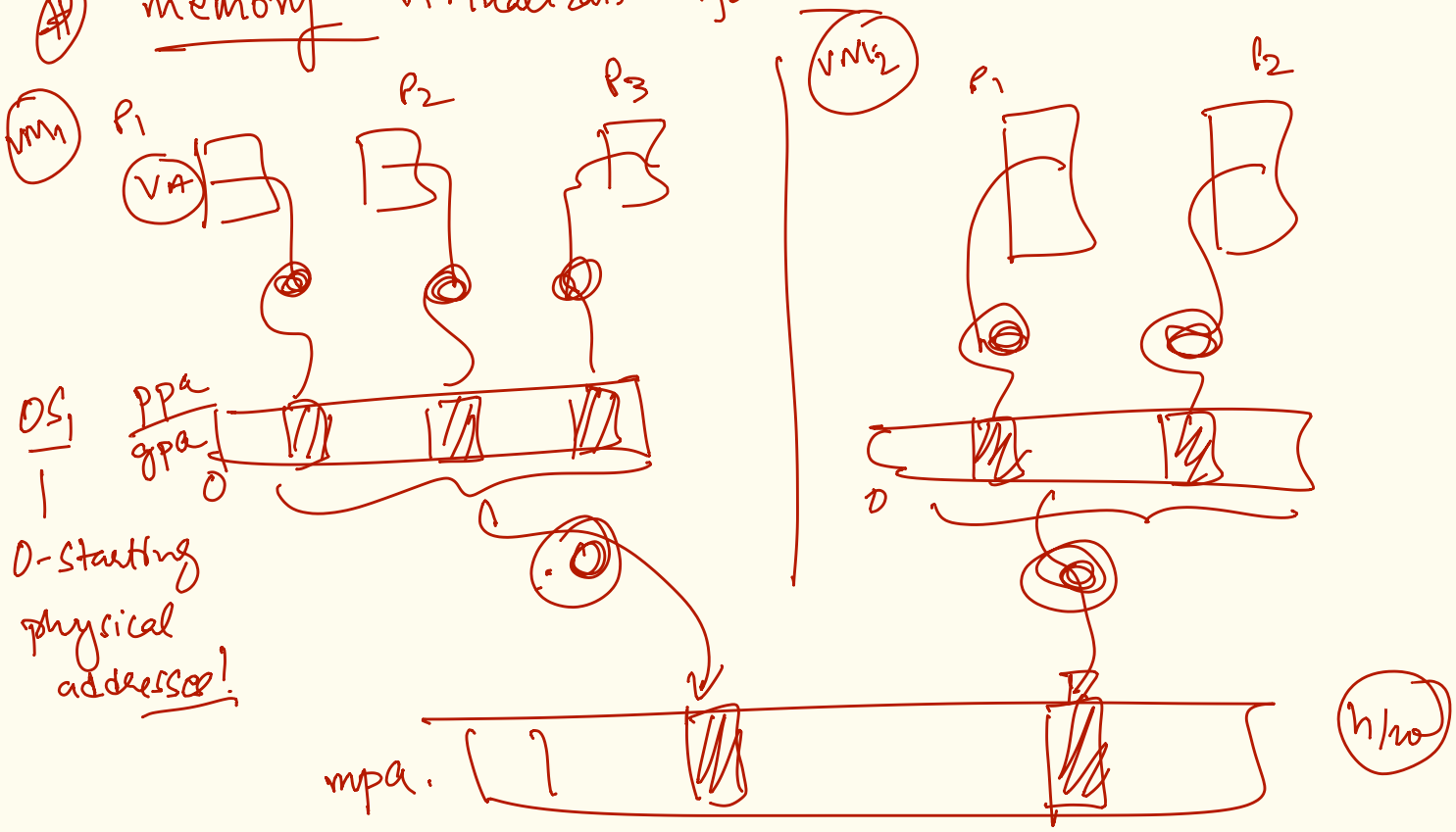
→ stores state of VMM on exit

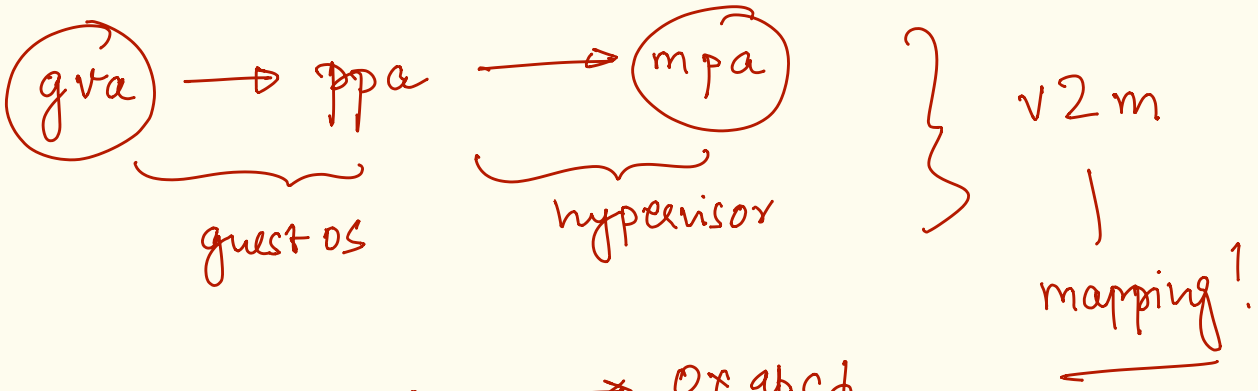
→ conditions of exit from guestOS

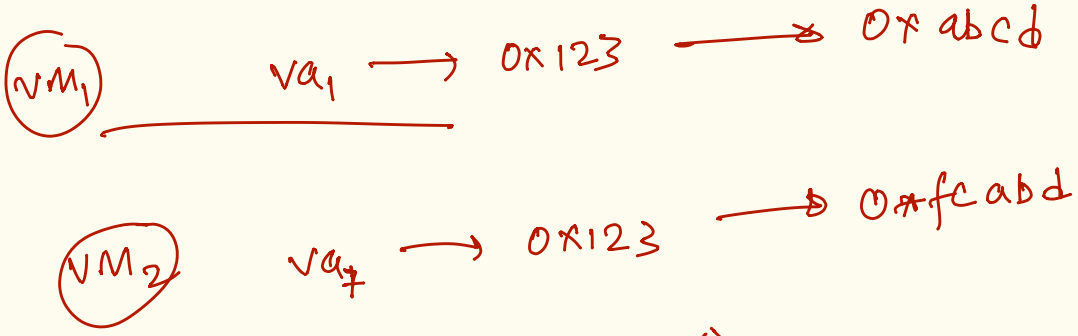
→ state to be store on cond.s of exit.

---

memory virtualization for VMs.







S/w

hijack all updates to pagetable

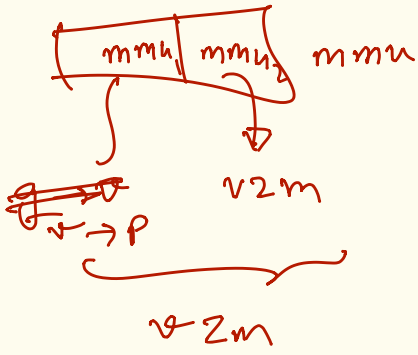
gva → ppa

ppa → mpa

update: gva → mpa in page table.

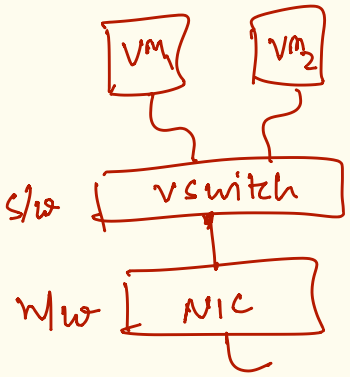
H/w

- mmu update to ISA which does nested page table walks

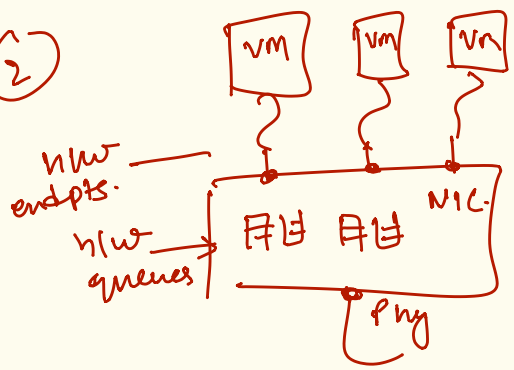


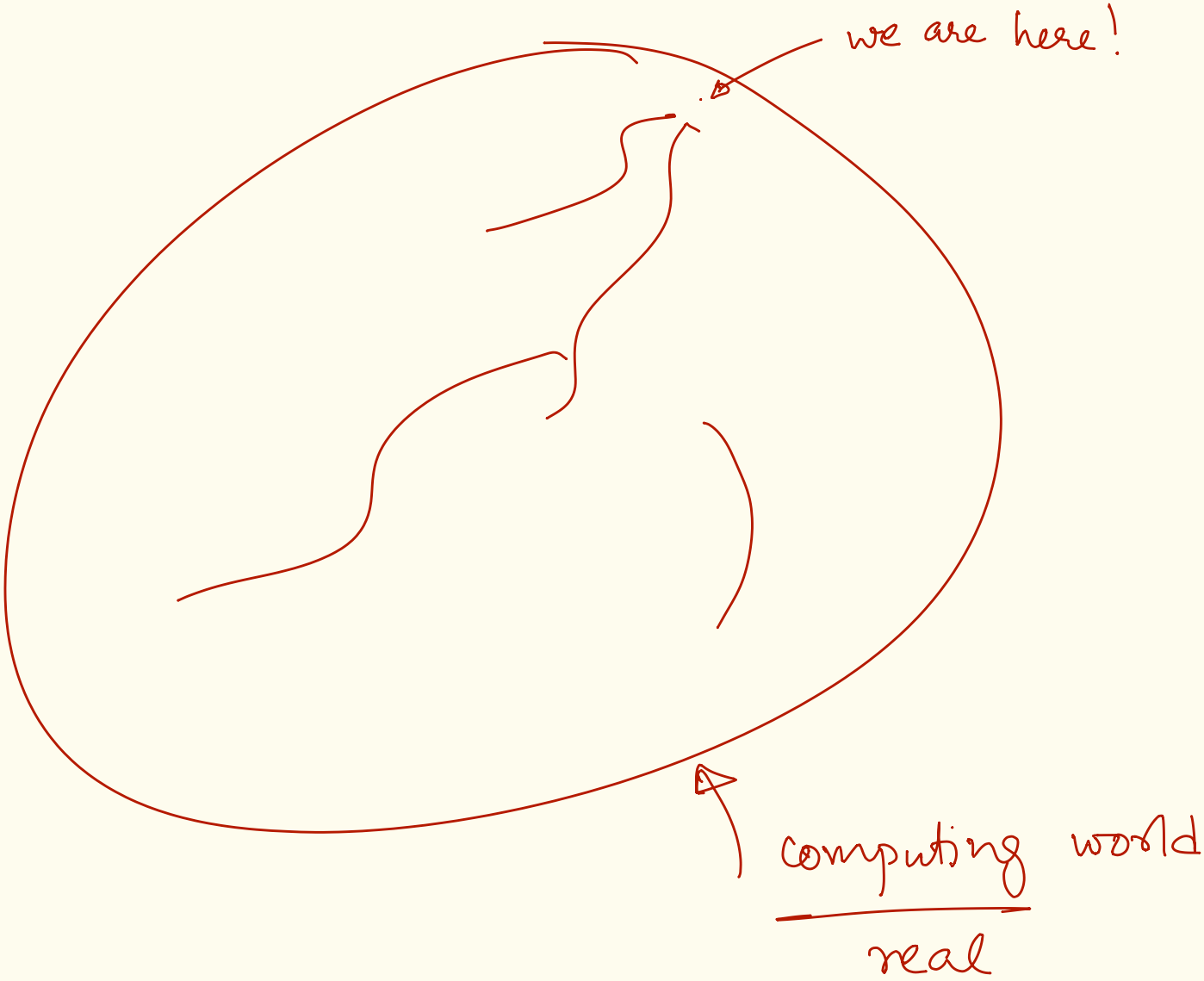
network

(1)



(2)




we are here!
computing world
real