Lecture #2 CS695. 11th Jan 2023

(*) process context, process the abstraction, OS the sole owner

process-view

- address space
- files
- device endpts.

⋮

process context

OS/system-view

- CPU
- ISA
- device specs.
- h/w support / privilege levels
- protection mechanisms for mem access

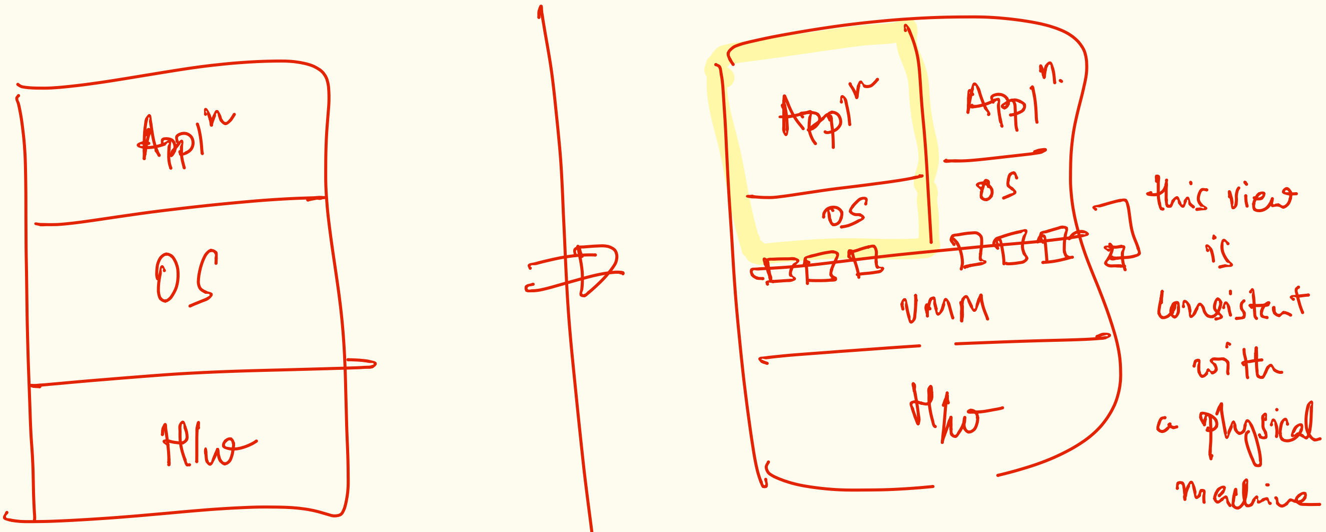
OS-context

process

decouples program from program execution.

(*) ⇒ new abstraction called the VM virtual machine.

~ env. to allow the OS-view.



this view is consistent with a physical machine

~ why VMs?

(i) building block for IaaS. < virtual compute infra. / on demand, flexible,

(ii) deeper isolation semantics.

(iii) multi ISA, multi OS runtimes on a single PM

(iv) debugging ~ OS!

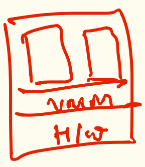
(v) new runtime execution features : migration, snapshot/checkpoint, record-replay.

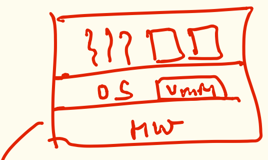
⤷ maintenance & upgrades.

(vi) VM image introspection

⊛ VMM design?

(i) categories of hypervisors.





- Bare-metal (Type 1) vs. Hosted (Type 2)
  - vmware esx
  - xen

  Hosted (Type 2)
  - kvm, vmware virtualbox

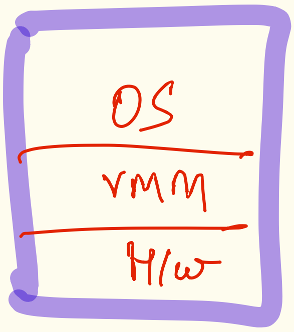
- Full-virtualization vs. Para-virtualization
  - vmware esx

  Para-virtualization
  - xen ~ citrix.xen
  - kvm

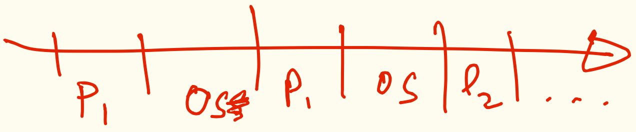
(ii) challenges of design?

IaaS
- compute
  - cpu
    memory
    IO

~~[illegible]~~ # ownership!
- cpu
- devices

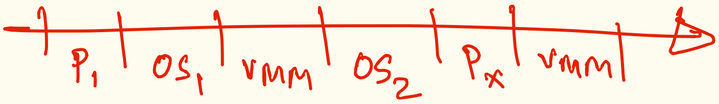


t cpu



~ 1 cpu shared by 3 diff. entities

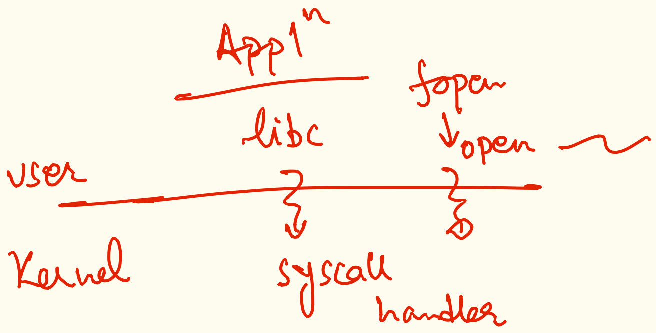
vmm, OS, processes



⊛ who is the owner?
- (guest) OS believes that it is the owner
- VMM has to be the owner.

e.g.: (a) update to CR3
- OS can write any value
- access other VMs memory!

(b) handling system calls. || ABI — ?



x86

%eax ← syscall no.

int 0x80 ← interrupt w/ vector 0x80

→ change in privilege mode user to kernel

- jmp to the interrupt handler

↑ IDTR

```
syscall handler {
    fn = syscalltable[eax];
    call fn;
    ret;
}
```

(iii) device virtualization.

└ devices cannot be chopped up (w/o breaking them physically)

⊛ Popek & Goldberg || 1974

principles of VMM design —

(i) efficiency ~ performance

(ii) resource control ~ ownership

(iii) equivalence ~ fidelity. ~ an OS/appl^n. should behave equivalently on a PM & VM.