

Lecture 7

CS 695

A1 — done

A2 — available later today.

⑧ Assignment #2 | virtual is real

~ simple-kvm ~ kvm-based hypervisor

↳ setup & execute VMs

userspace program
hypervisor

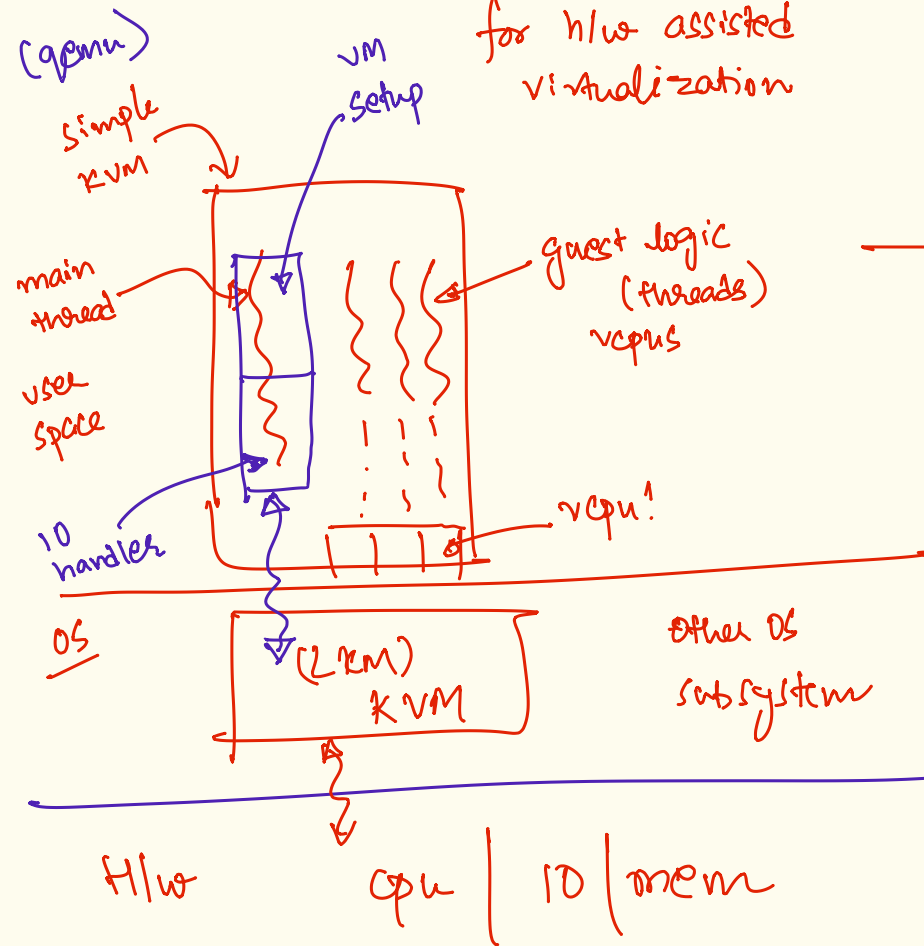
⊕
KVM — linux LKM

for h/w assisted virtualization

simplified guest logic
single binary

- No: guestOS + App

- Yes: single program in a VM.



```
devfd = open("/dev/kvm");
```

```
vmfd = ioctl(devfd,
             KVM_CREATE_VM);
```

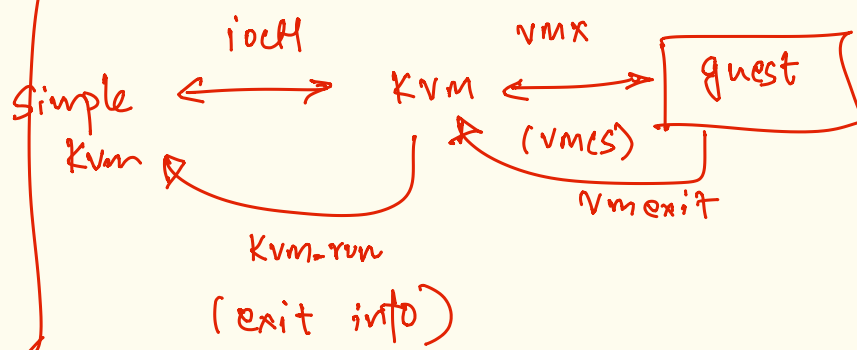
```
vcpu = ioctl(vmfd,
             KVM_CREATE_VCPU);
```

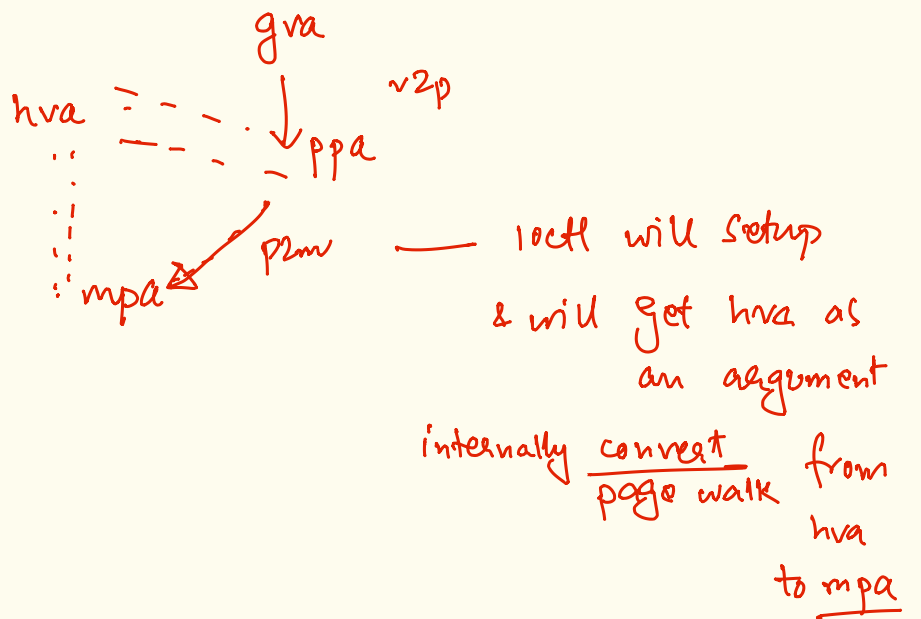
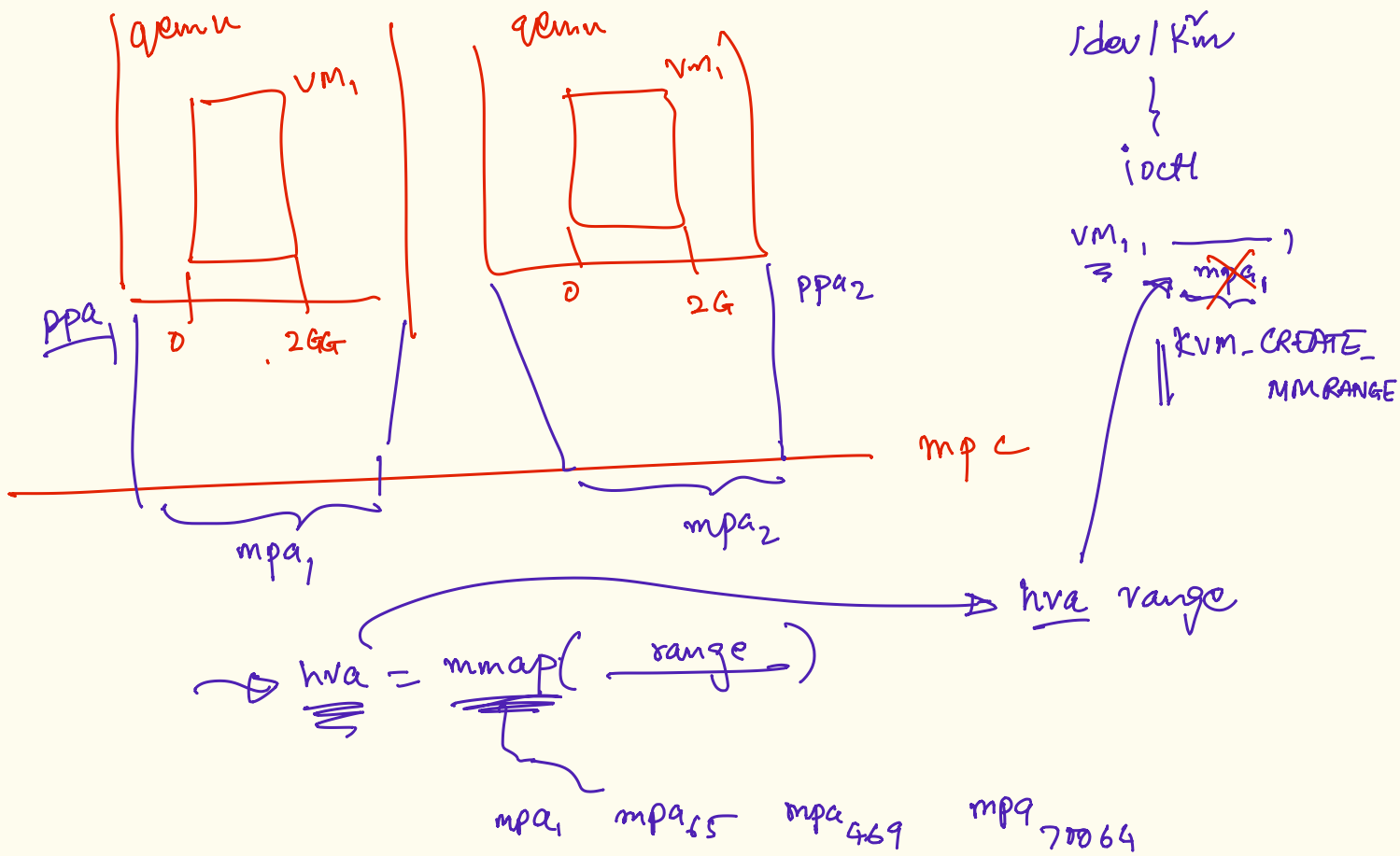
```
vcpu/vmfd → mem
```

```
for (all cpu)
```

```
r = ioctl(vcpu, KVM_RUN);
case(r): // exit reason
```

gva — ppa — mpa
hva

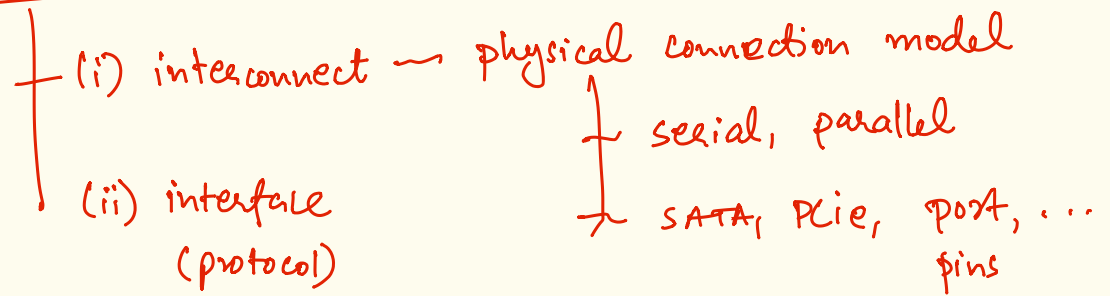




IO scheduling for VMs via the hypervisor.

~ OS/system-view (VM) consists of IO devices

~ devices have



~ pio ~ programmed IO ~ ISA instr. in/out for IO

mmio ~ memory mapped IO ~ R/W on memory addresses

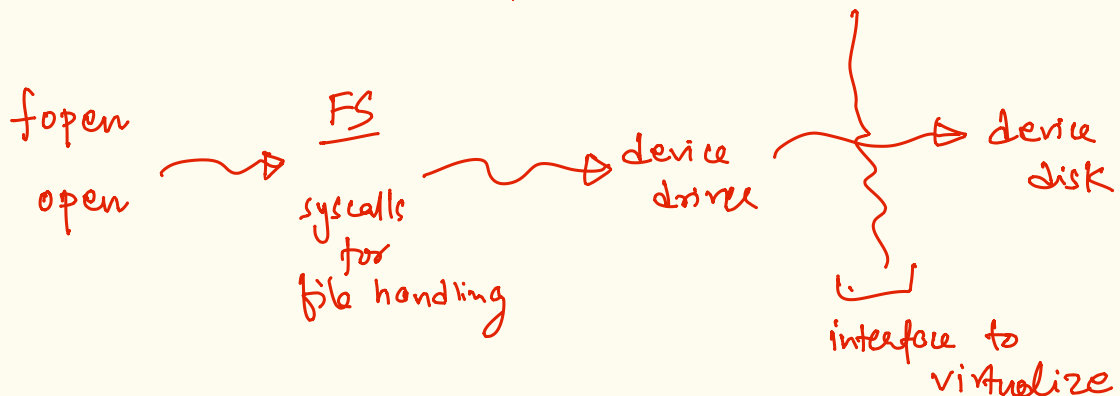
~ dma ~ block copy from/to CPU - device
device registers

Example

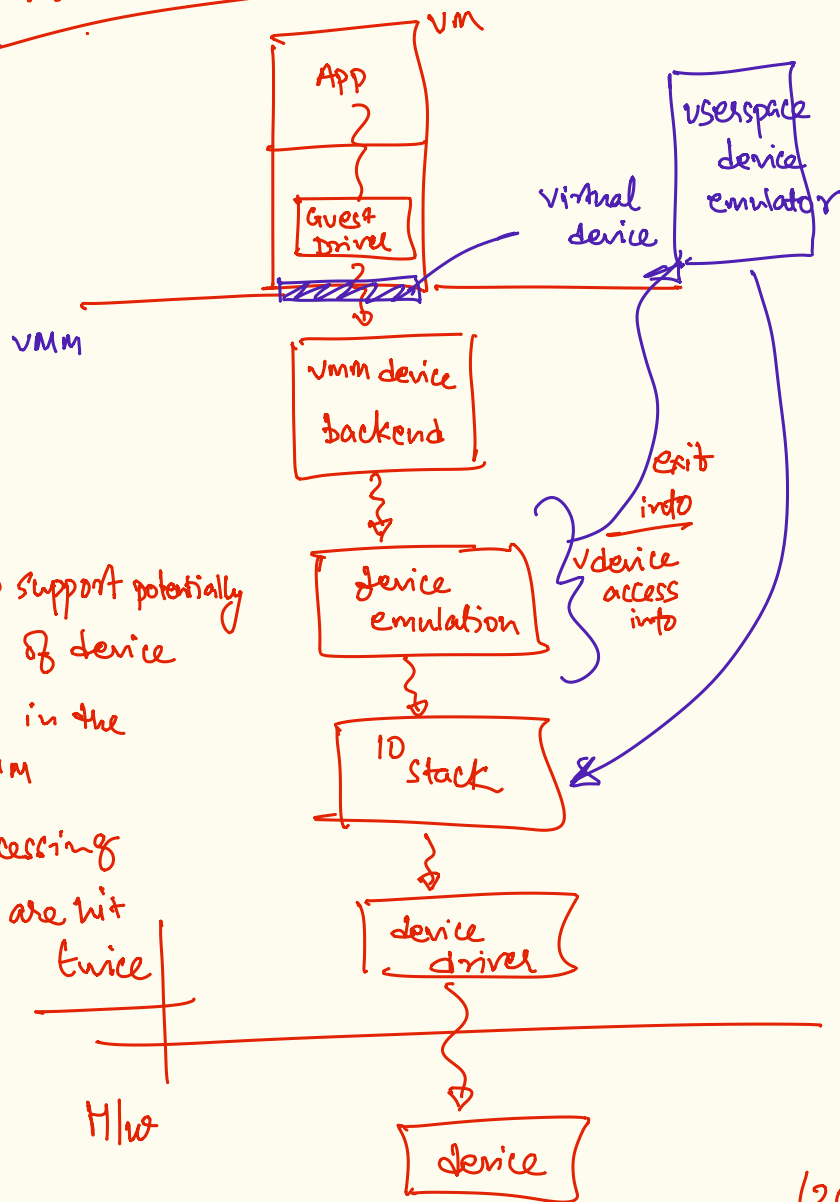
Disk IO

- assume the following interface for the disk

Reg. 1 ~ ~~sector~~ ^{block} number
2 ~ R/W
3 ~ memory address
cmd ~ in/out



① Full virtualization



Cons:

- need to support potentially large set of device interfaces in the vmm
- io processing costs are hit twice

Reg. 1 — 23
 2 — 0
 3 — %addr
 in

↑ read block 23
 & copy to %addr

23, 0, %addr, in | guest

read from disk on local m/c at real block number (2023, 0, %maddr, in)
 read(offset, fd) → + (23)
 nfsread (network)