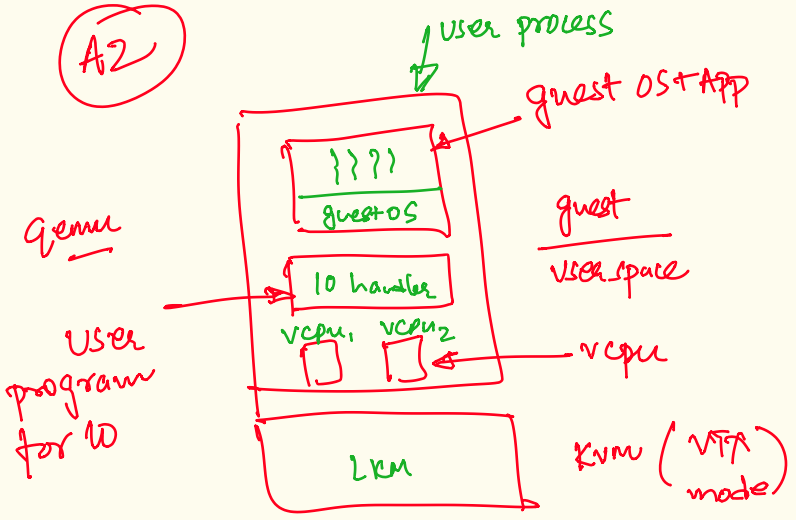
Lecture 8 — CS695

A2




```
vmfd = open(/dev/kvm)
vmcpu_fd = ioctl(vmfd, KVM, CREATE_CPU);
for(all cpus)
status = ioctl(vmcpu_fd, KVM_RUN);
    case(status) { //EXIT reason;
        IO :
        default:
    }
```

assignment

(*) A2 — 19th Feb. 11.59.59 pm

(*) Quiz 1

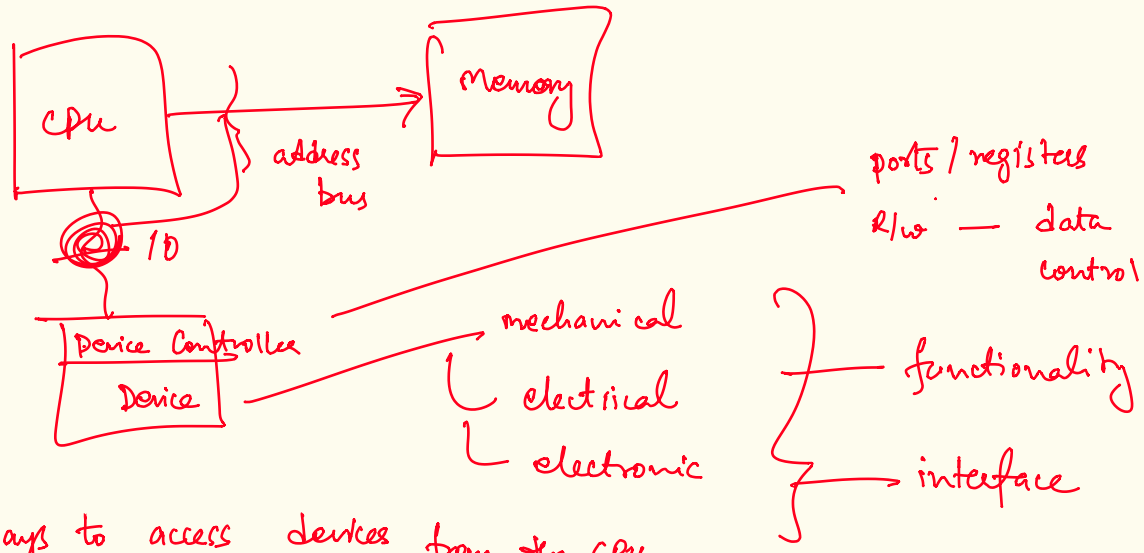
7th Feb. 830 am. Wed.

attempts to modify first law w.r.t. AI

pros	cons
new · output at very energy no energy transfer	-
- time to do other stuff	-
- free lunch!	
- get to potentially meet the Dean on a personal appointment	

IO virtualization

access to/from devices other than the CPU



(K) ways to access devices from the CPU

(i) PIO — programmed IO — ISA instr. to be issued by the CPU for IO

(ii) MMIO — memory mapped IO
- map memory regions to IO regs / ports
- memory interface ⇒ IO access

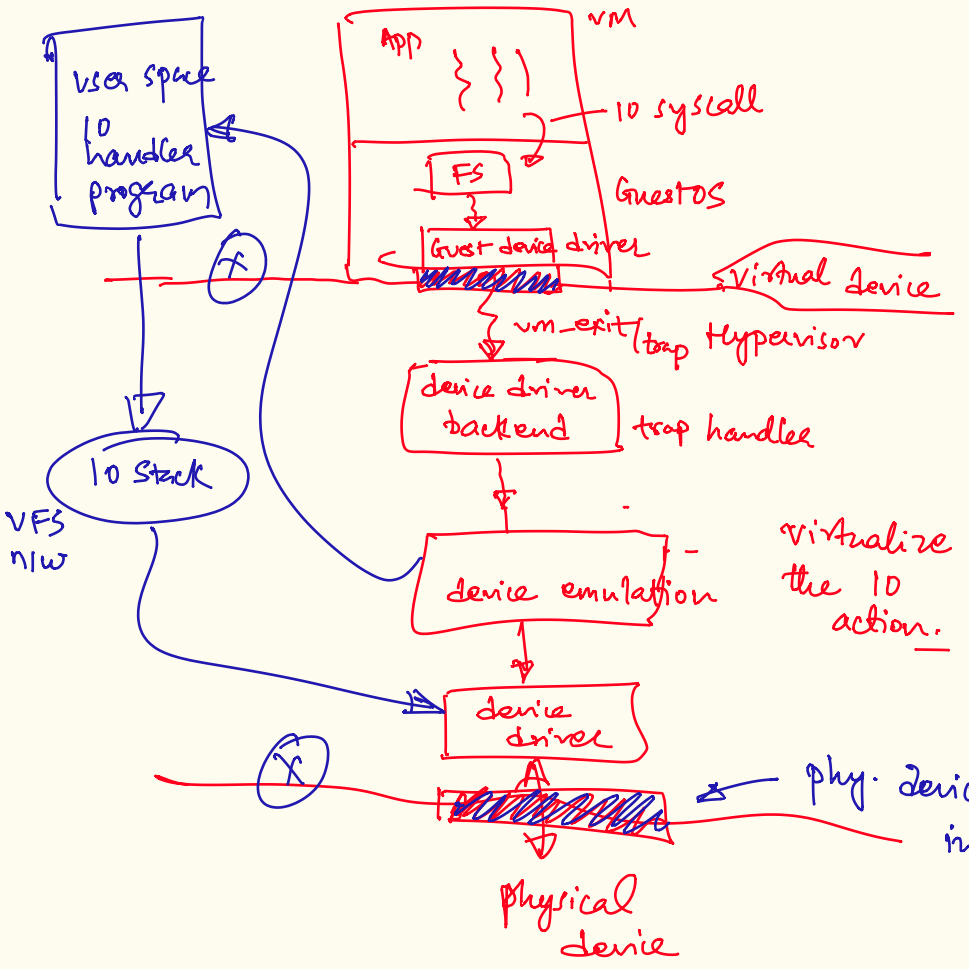
(iii) DMA
direct memory access.
for copying data to/from memory to device mem/regs.

e.g. disk interface

Spec. & such an interface.

~~reg.~~ x	← block number	100
y	← R/w	0
z	← address (virtual)	0xa10000
in/out	← instruction	!

Generalized IO virtualization view



x ← 100
y ← 0
z ← 0xa1000
in/out

VM_1	VM_2
x – 100	x – 100
y – 0	y – 0
z – 0xa1000	z – 0xa1000
in	in
z ← 0xb1000	0xd1000
y ← 0	y ← 0
x ← 1100	x = 8100
dev. id ←	dev. id ←

guest: 100, 0, 0xa1000 } vdev interface → VMM: file = fread(fd, 100×512, buf); copy(buf, dest);

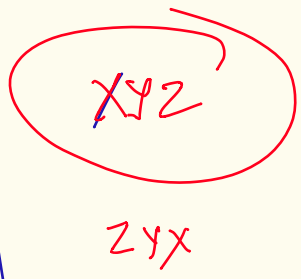
guest driver | vm-exit + kvm handling | user space

① Emulated IO
(Full virtualization)

- VMM emulates the full device specification.
- 1 : 1 guest device (VMM) phy device
- needs to understand & implement / virtualization each action, config. state setting via the device interface.

② PV IO

Split-device driver model.



③ HW assisted IO multiplexing virtualization.