Lecture #2   CS347   4/8/2023

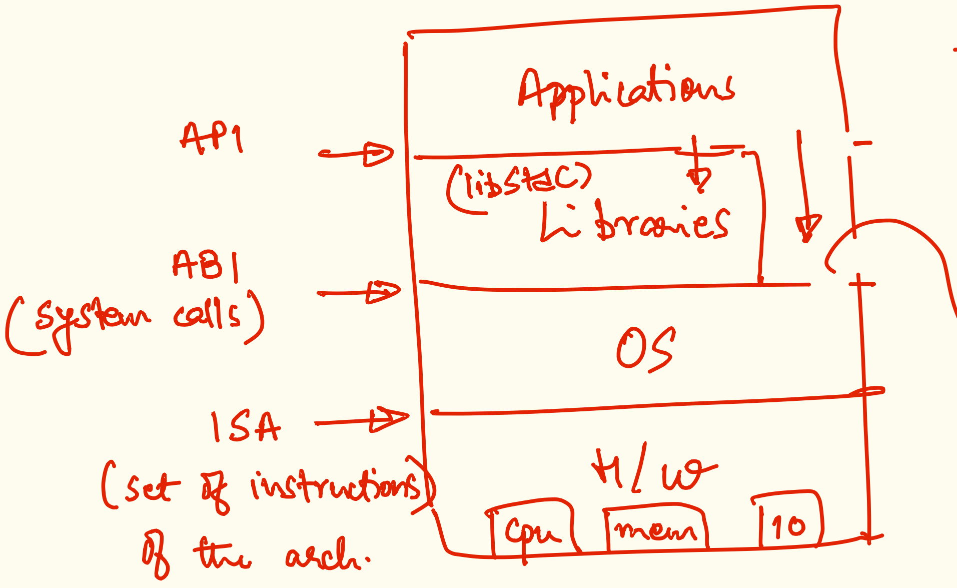
recap: - OS essential for world peace!

- abstractions & interfaces

(i)

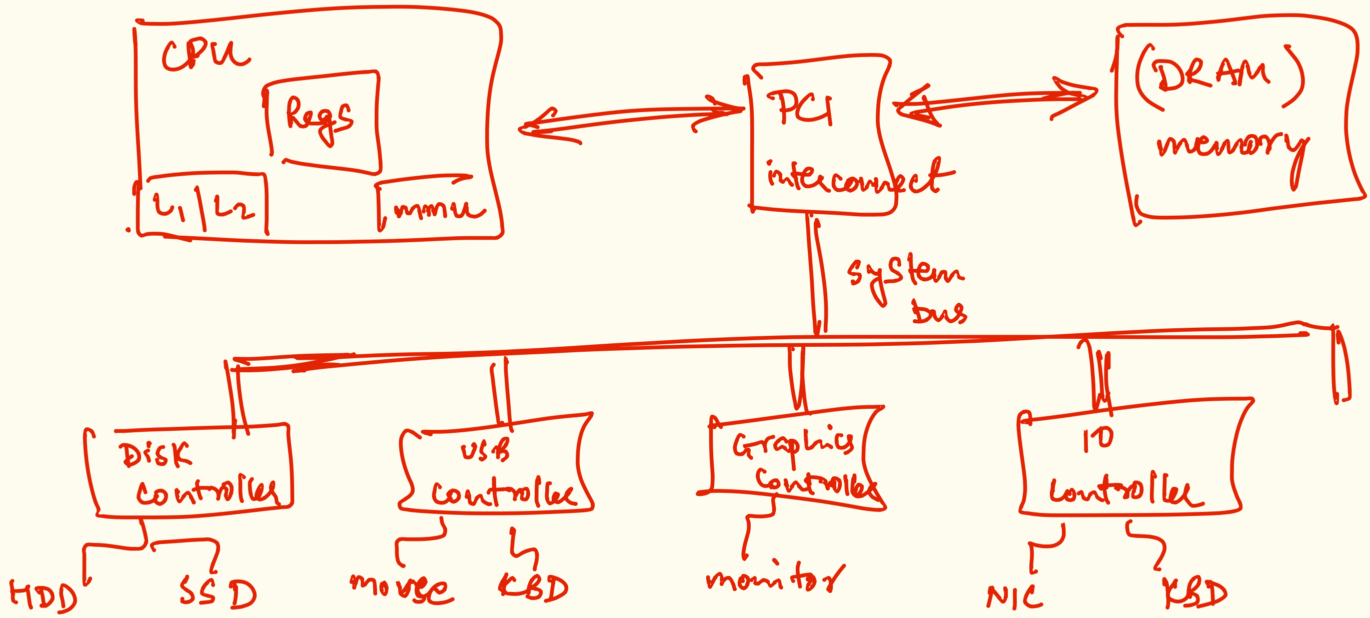
HW

- API vs ABI ?



fork
open
close
wait
execv

⊛ simplified architecture view of a machine.

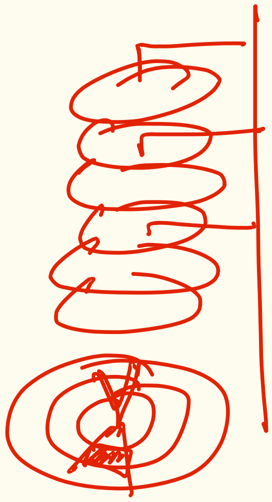


- compute resides in the CPU & the controllers.

- disk controller:
  + providing a linear blocks interface.

  HDD ⇒ 'n' blocks

  - hides. block # to sector, tracking + disk specific signalling.



- CPU ~ von Neumann model.

  fetch-decode-execute

```
label: fetch from current pc/ip
       decode
       execute  ⟹  pc may get updated
       next PC
       jmp label
```

(ii) OS requirements

- multiple execution instances

⊛ └ efficient use of capacity / min. overheads !

- user mgmt.
- disk partitioning/provisioning
- useful abstractions

⊛ – correctness

- error handling/robustness

⊛ – isolation

- extendable / H/w independent ! (?)

↕ list of bad things.

- peek into others memory.
- monopolize the CPU !.
- suck up all network pkts.

— interval ~~in~~ timer ← # CPU cycles ~~before~~ after which OS intervenes!

↑ register on the CPU

x all critical setup / configuration / allocation has to be via the OS!

design principle.

(i) privileged mode of execution (LDE limited direct execution)

(ii) interrupts / interrupt-driven execution.

(i) privileged mode of execution

every instruction of the ISA has a min. privilege level for correct execution,

the OS sets CPL ~ current privilege level

for all execution.

---

x86. Ring 3 —— Applications. user mode

2

1

0 —— OS

Kernel mode