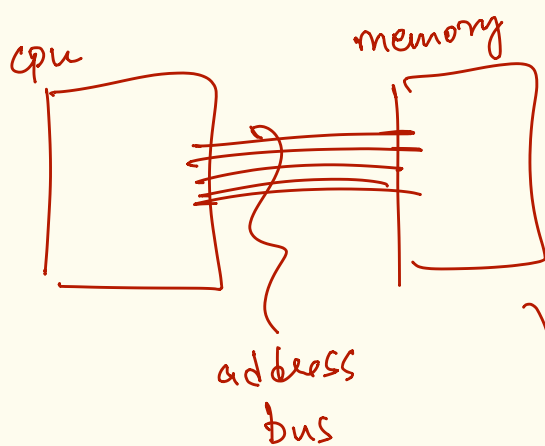


von Neumann model (Princeton)
(x86, arm, ...)



vs. harvard model.

(µcs, L1 cache, DSP...)

- Lab 1
- Lab 2 (out today)
- No Audit
- Lab 3 (out next week)
- Lab session (soon)
- TA office hours

- shared / common memory } accessed via
for data & instructions } the same bus.

- pointer / reference to data / memory }
PC / eip

all cpus of the world:

while (1) {

fetch instⁿ. from PC

PC++

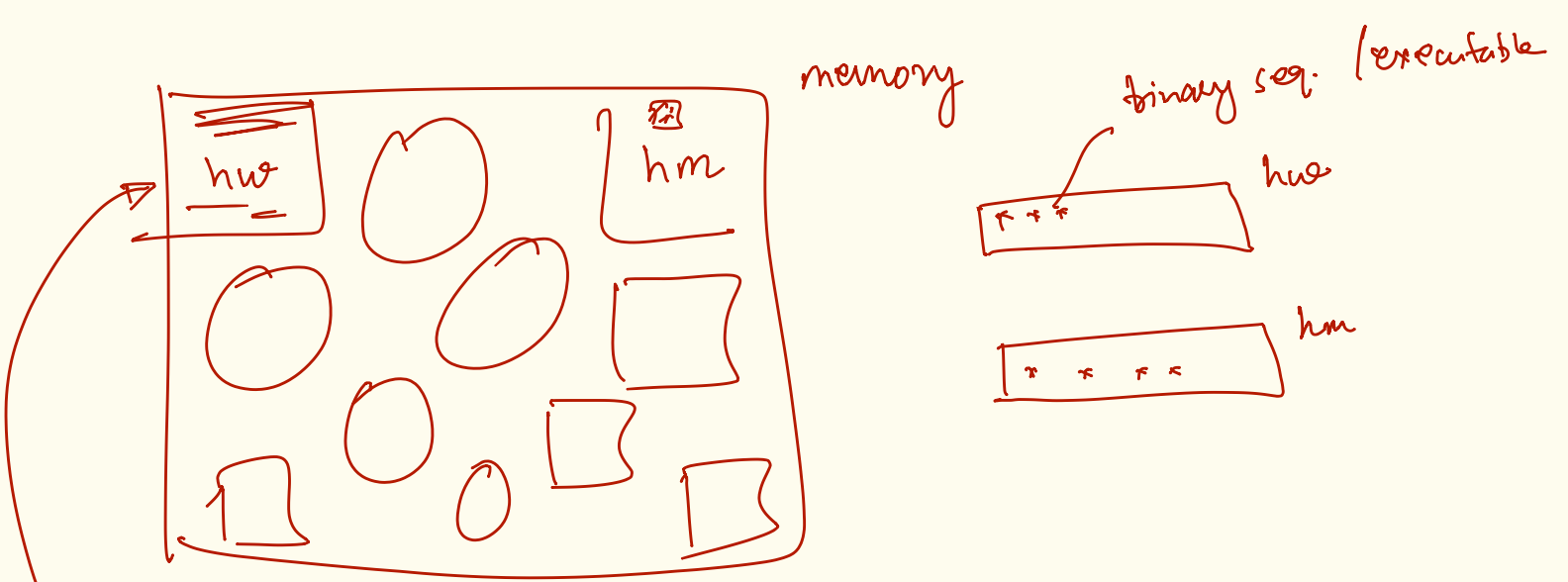
execute instruction

}

// side effect
of changing
PC,

What is an Operating System?
why

Design &
Implementation
highlights



OS is a

(*) slow entity that provides useful abstractions (& interfaces) for general purpose computing and enforces efficient, safety, isolation of hardware resources.

(*) what will happen w/ no OS?
bad things

- inconvenience / large developer overhead.
- programs can peek into each others memory!
- monopolization of resources (CPU)
- all n/w packets gobbled by same app!?
- safety! — one program crashes! will it effect every other program?

...

on some ISAs

time interval register

- stores the number of clock ticks before an event/interrupt
- used to give time slice on CPU for appl's.

Q ^{who} ~~how~~ / which entity writes to the register? $\rightarrow$ not the program.

— But the co-ordinates
OS.

④ one-rule for the OS design model.

OS is the owner / sole controller
co-ordinator
of all hardware resources!

```
mov    eax, 23
```

```
mov    esp, 24
```

to establish control

OS depends on the ISA!

① Every instruction of the ISA has associated minimum privilege level (of the CPU) for correct functionality.

② the CPU at all times is in some privileged mode level.

③ critical instructions (are a subset of ISA)
that manipulate runtime state / behavior
which usually need higher privileges.

