CS 219 Lecture #4.

Lab 2 out soon!

~ OS requirements!

what are possible bad things that we want to avoid?

- ~ processes/programs to peek/overwrite each others memory.
- ~ take down every entity along with you
- ~ monopolize the resources (CPU)
- ~ suck up all network & disk IO to a single entity (program)

⊛ interval-timer register ~ on some ISAs

- a register to store a count-down counter
- generates an interrupt on zero
  - stops work on the CPU
  - provides an opportunity to do something else.

Q. which entity writes to the register?

↳ OS!

↳ OS is the sole owner / controller of the hardware resources on which it builds its abstractions.

↳ all critical / sensitive configuration has to be OS-done / vetted.

(*) two basic design principles

(i) ~ privileged modes of CPU — LDE (limited direct execution)
execution

(ii) ~ interrupt-driven execution.

(i) LDE — execute on the CPU in a limited manner

till no critical instructions executed.

~ LDE is ~~built~~ built on the feature of ISAs which state the following —

① # CPU can execute in a privilege level

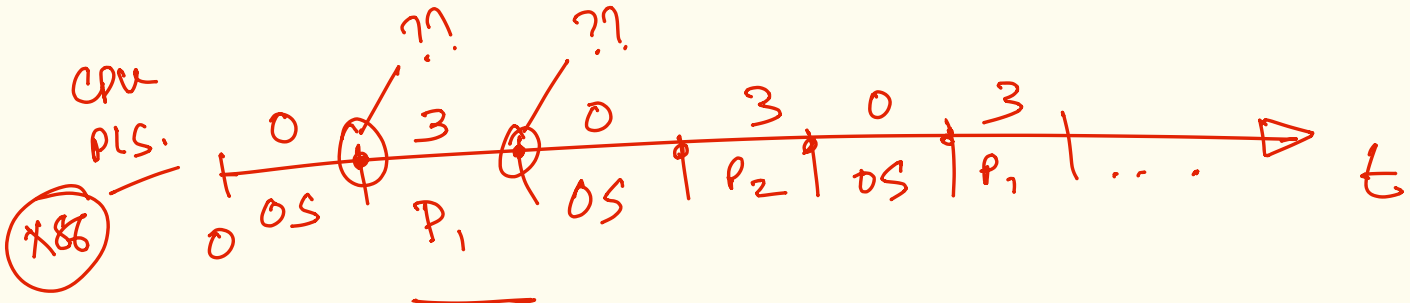
least
↳ Ring 3
2
1
0
most

x86

risc-V
U — user
S — supervisor
M — machine

② every instruction of the ISA has a stated/specified minimum privilege level for correct execution.

↳ ⁂ all critical instructions require the highest privilege levels/modes.



↳ cannot do/execute critical instructions.

(ii) interrupt - driven execution.

~ to invoke (implicitly the OS explicitly)

~ all IO non-deterministic about when.