CS 219 Lecture 5

# OS building blocks

1. privileged modes - based execution (LDE)
2. interrupt - driven execution

meta-invariant: OS is the sole owner / controller of all resources.

~ stop a sequence

~ stop a sequence of instructions on the CPU.

~ next inst^r. after interrupt not from PC+1

PC changed to inst^r. seq. that "handles" interrupts

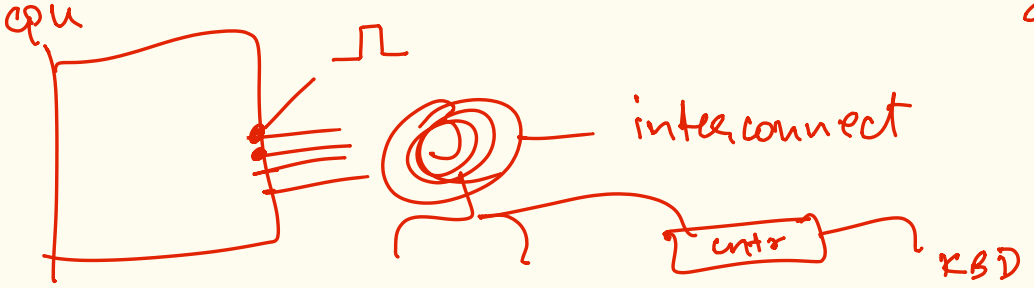
~ "state" / context of sequence has to be stored (to be restored later)

(Stored somewhere - where-?)

~

(#)

- keep executing instructions
- on interrupt, pause,
- ~ handle interrupt
- return to loc^n. where seq. was abandoned.



# what is an interrupt ?

- physical pins on the CPU connected / toggled via an interconnect to signal interrupt & device information.

---

program

```
mov  R0, 23
add  R0, R1
sub  R3, R4
inc  R6
```

PC

interrupt!

context

- save context
  - save all registers.
    - general purpose
    - special regs
- restore context & execute next instruction.

first OS abstraction

CPU virtualization

- share / multiplex the CPU
- decouple program development from program execution.

via the process abstraction

## program vs process

**program**

source code — preprocessor
|
compiler
|
linker
|
(binary) file

- format / template
- ~ ELF : executable linking file format

**process**

- ~ OS abstraction
- program in execution
- "instance" of a program
- entity that consumes resources
- set of instructions & data loaded in memory
- decouples execution from program
  - # instances
  - priority
  - scheduling, resources
  - start, stop, pause