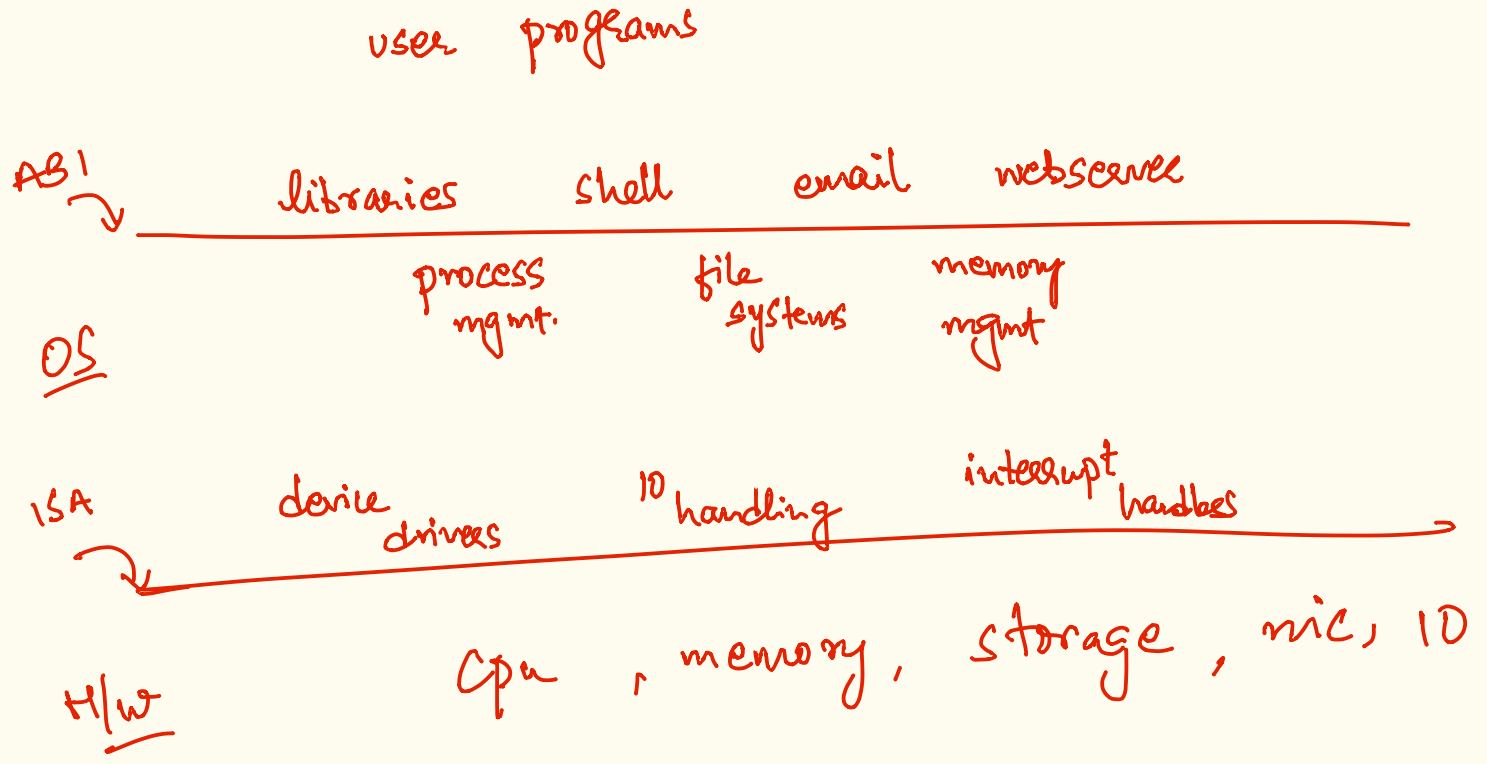


recap: OS is/has to be owner of all (managed) resources.

for correct/reliable services & abstractions.



① two-building blocks for the OS design.

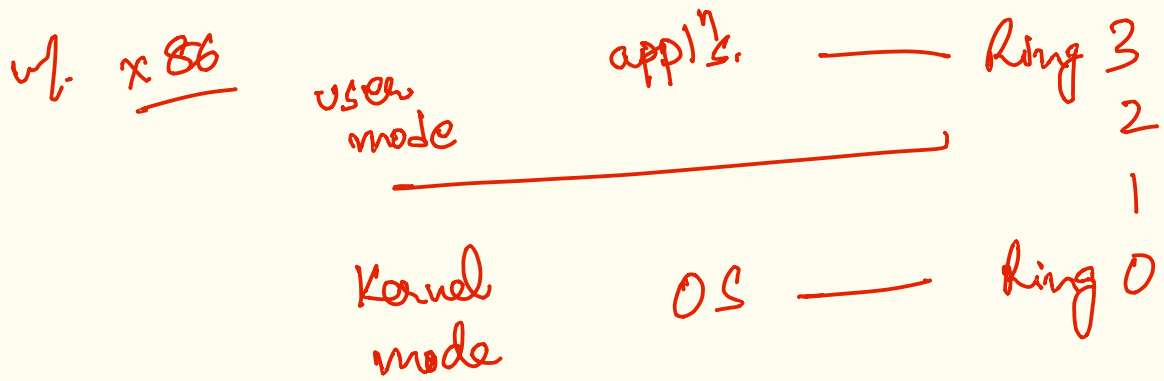
- (i) privileged modes of executions
 - (ii) interrupt-driven execution
- LDE
limited direct execution model

H/w { CPU executes with apl some privileged level at all time.

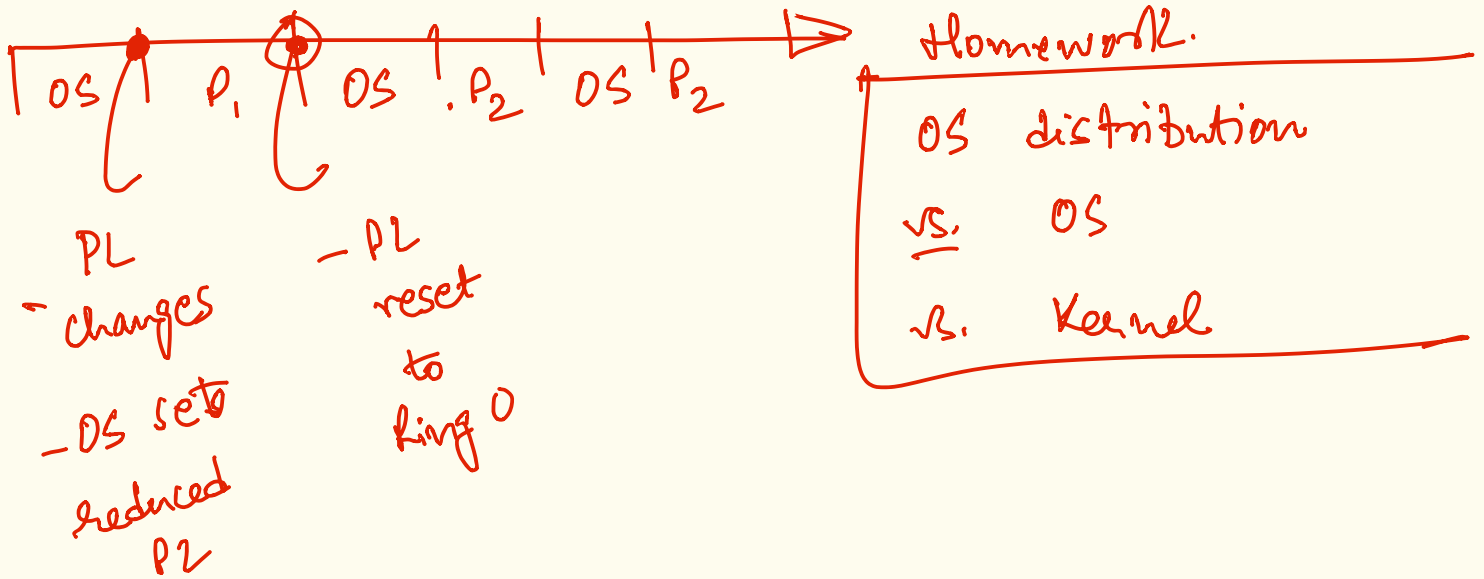
PLs are - configurable. eg: w/ x86 last 2-bits of code selector which represents the PL.

~ each instruction of the ISA has an associated min privilege level for correct execution.

e.g. SET interval timer cmd. — PL — ring 0



time



(ii) interrupt-driven execution.

what is an interrupt? (H/w assisted)

- event to stop/pause CPU ~~to~~
- change PL to Ring 0 // x86 specific

* how is an interrupt generated?

- change in a voltage level on a CPU pin/pins

- int / INT

Interrupt number → x86 ISA instruction.

⑧ why interrupts?

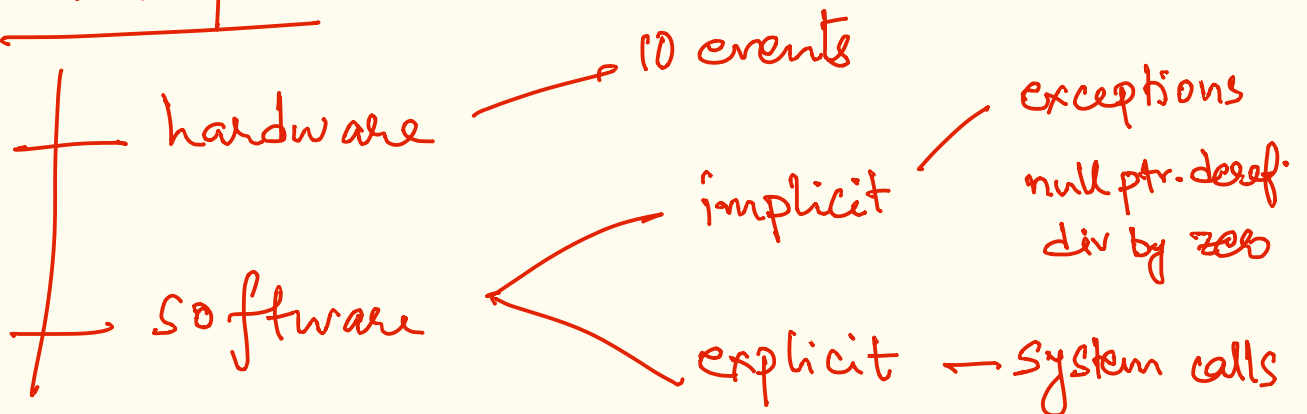
world is non-deterministic

all IO events are non-deterministic

↳ key press, disk read completion, arrival of new pkts.

used as a building block for the ABI / system call interface.

interrupts



(*)

program

vs

process

source code

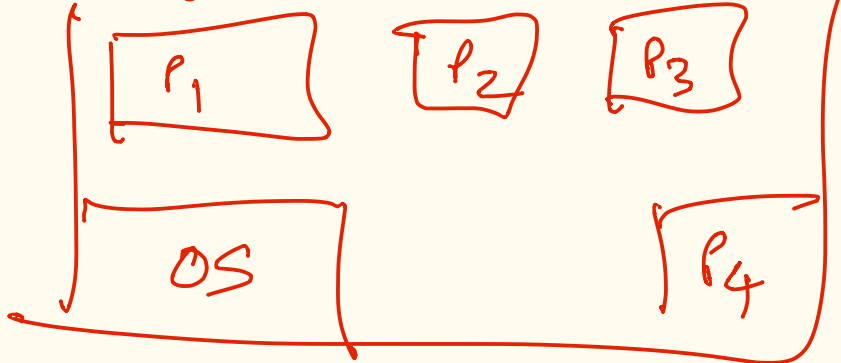
libraries



executable binary program

- software paper-weight + hardware
- ELF format

memory



~ as an instance of a program in execution.

~ base unit to deliver & manage services.

duplicates a process - fork

exec

loads a program into process

metadata of each process - PCB
process control block

- + pid, ppid, state
- + memory alloc info. / bounds
- + open files

- + ^{cpu} registers
- + kernel stack