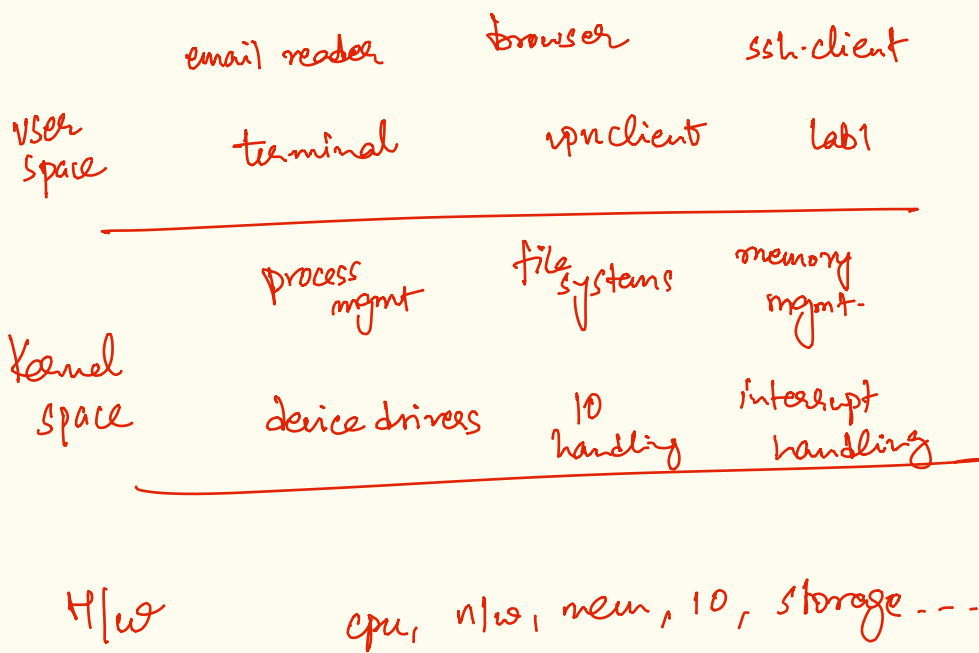


Lecture #4

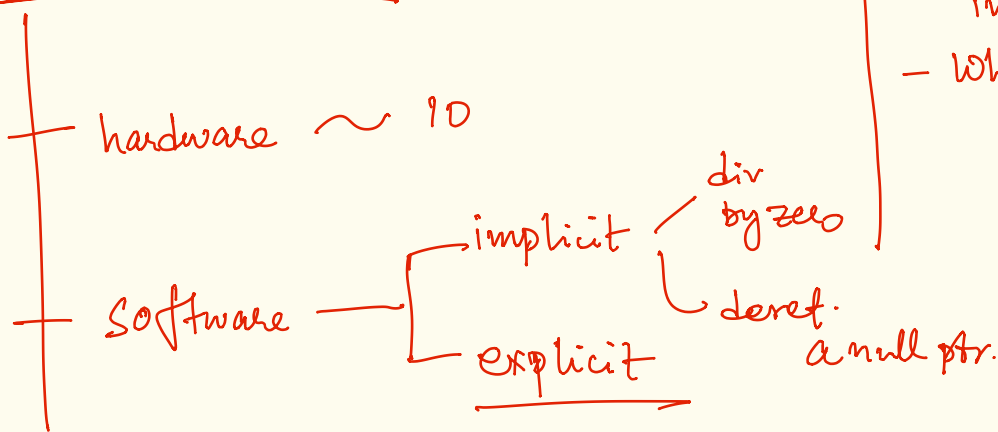
CS 347m



two building blocks -

(i) LDE

(ii) interrupt-driven execution.



x86. instruction INT <interrupt> vector

INT 0x80

① Lab1 - TA hours
- 11.15 am today!
- ??

② Lab2 ~ later today!

③ H.W.

- API vs ABI

- Ubuntu vs Debian
vs. Linux

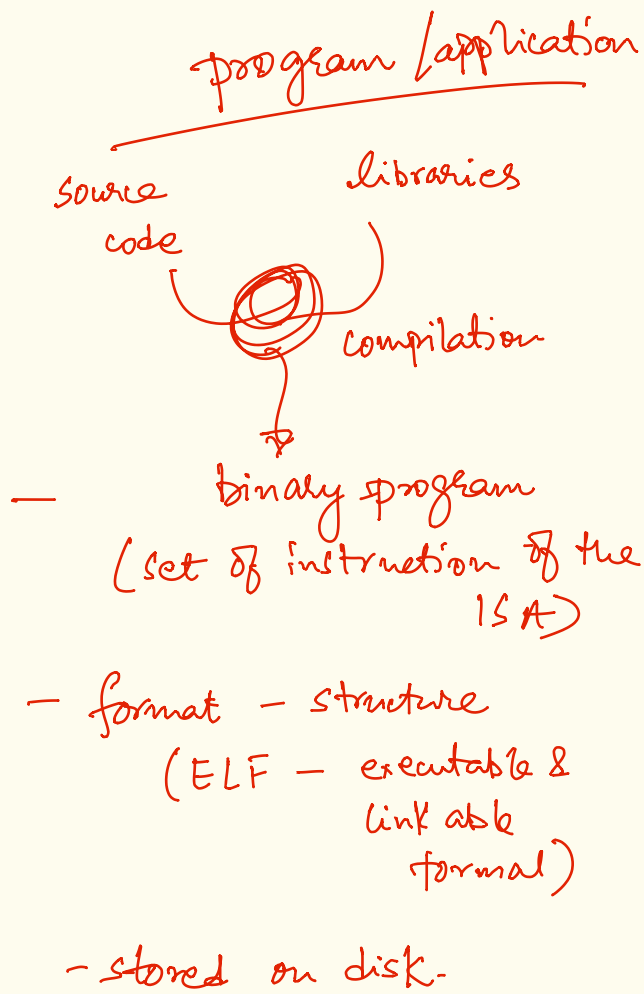
- List of all processes on your system

- Some process names are in the format [kswapd] in brackets.

- What do they indicate?

① efficient CPU usage / sharing.

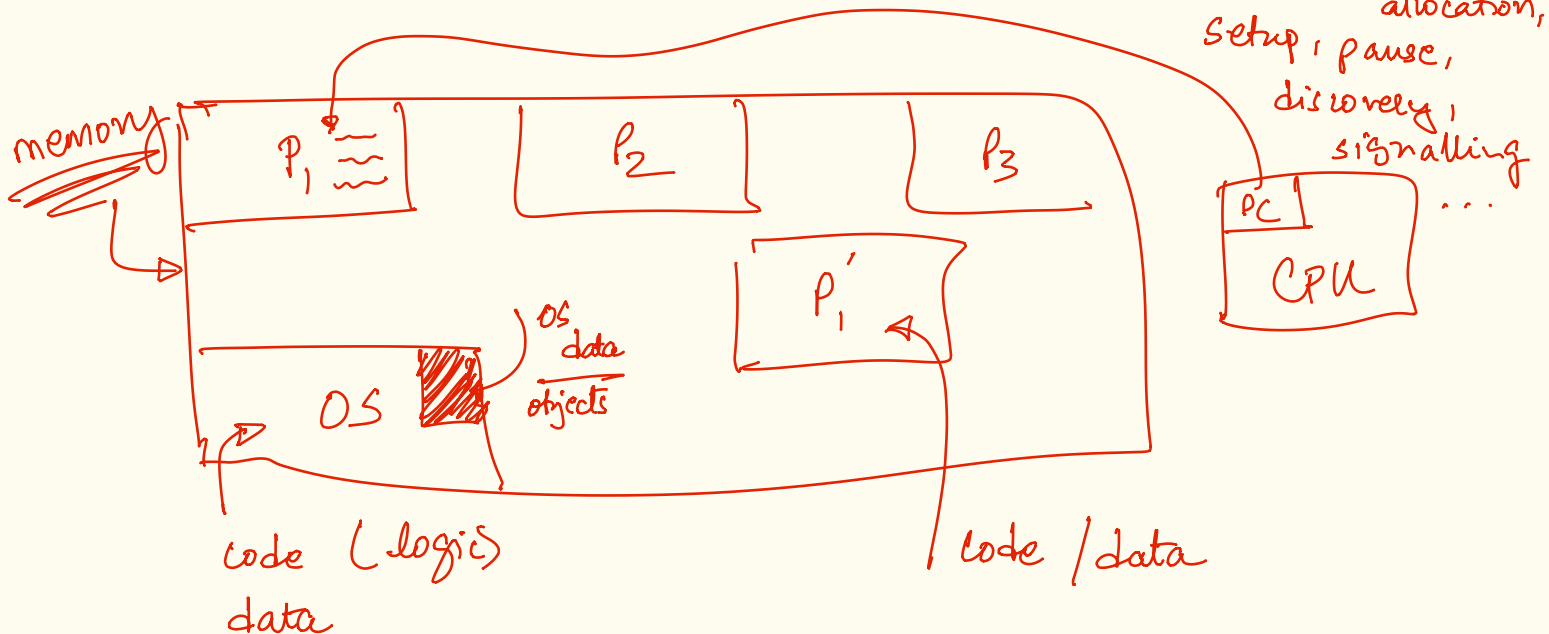
~ process abstraction



vs.

process

- ~ OS abstraction
- ~ program in execution
Instance of a program
- ~ entity which consumes / allocate to hardware resources
- ~ set of instructions loaded in a memory region & pointed to by the CPU.
- ~ enabler for decoupling program logic encoding from its execution.



② for every abstraction the OS has to store meta-data / information regarding the abstraction.

eg ~ cab-hailing service : #cabs, #drivers, billing, traffic
#free cabs, location-based info, ...

process — { program in memory (memory region)
~~data~~ current PC

Q is the process always executing on the CPU?

— NO!

~ waiting it's turn (to be scheduled on/ allocated the CPU)

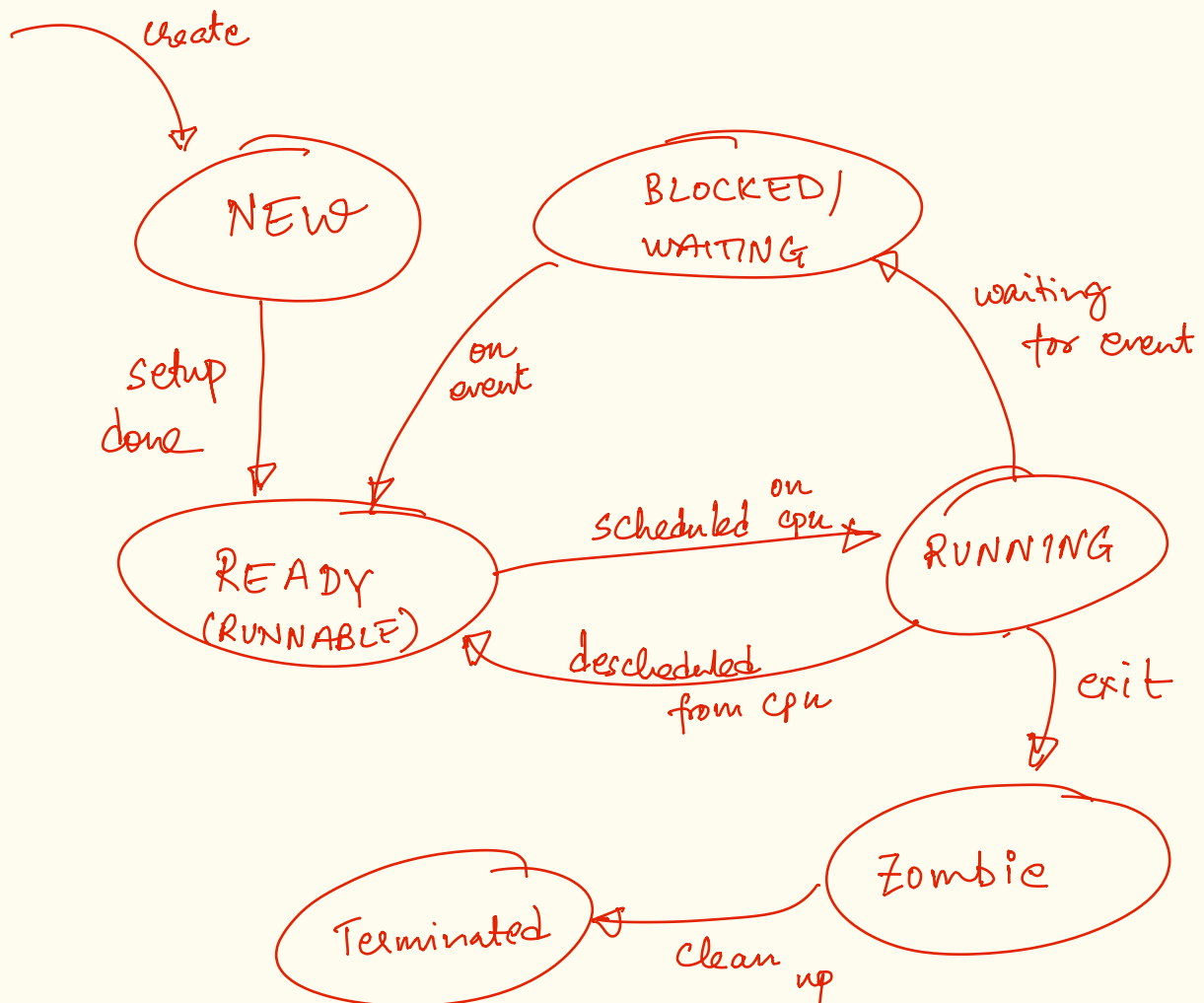
~ waiting for an event
↳ (IO, condition, ...)

~ done execution

~ being setup/loaded

process life cycle

process state diagram.



⑧ process abstraction's metadata is stored
in PCB ~ per process control block

└ metadata object / structure

├ pid, ppid

├ PC, CPU context

├ state

├ memory region information

├ priority

├ open files

└ signals ...