

Lecture #1

1/8/2023

- ① systems / operating systems
- ~ to a set of concepts / ideas / engineering O/S.
- that enable / capabilities to develop programs
build applications
- ~ DBs, Arch, Compilers, assemblers
OS IDEs linkers, networking

② what is an OS?

- "manages" all programs
- ~ connects h/w and s/w components
- ~ provides interfaces between users & h/w.
- ~ provide security / isolation / for programs // safe execution
- ~ s/w to build applications. ^{failure detection}
- ~ enabler for mass / ubiquitous computing.

③ what are common actions / services of an OS?

- virtualizing.
- ~ start / stop / pause / ... programs.
- ~ scheduling
- ~ alloc / dealloc of memory
- ~ Tx / Rx pkts.
- ~ R/w to a disk / file.
- ~ IO to external devices.

② Why an OS?

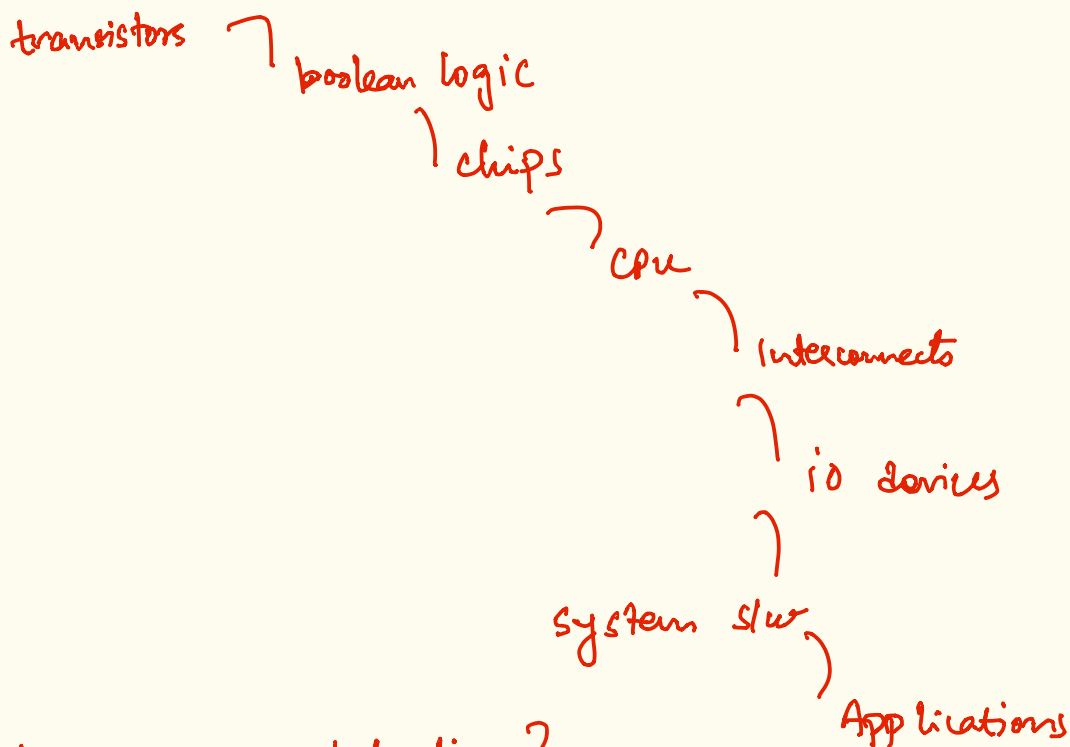
— mgmt. layer needed for orchestration.

~ h/w independent source code

~ programs are buggy!

~ abstractions over h/w

~ what is a machine? (architecture sense)



⊗ what is an abstraction?

↳ conceptual functionality to be consumed
used

↳ details not the concern of user

↳ functionality accessed via an interface
consumed

⊗ what are common OS abstractions?

- files, sockets, process, io endpoints

...