The Abstraction: The Process

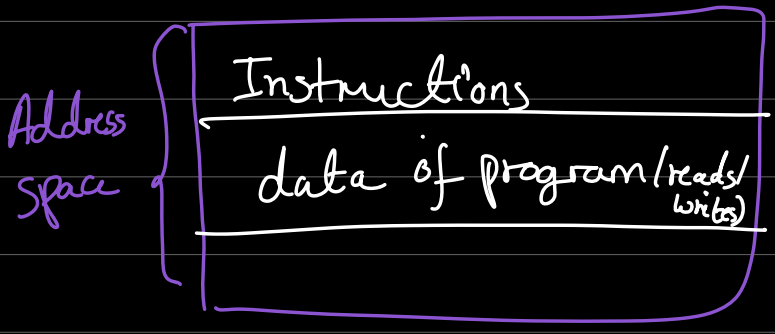
Virtualizing the CPU

- Time sharing - Each process gets time on the CPU, getting run/stopped multiple times
  Context-switch - OS gets the ability to stop running one process and start running another on CPU ⇒ time-sharing

Among multiple processes which to run at a specific time ⇒ scheduling policy

The Abstraction: A Process

- Machine state



Note: Diagram just to depict things part of process

- Special registers ⇒ Program counter (PC)
(instruction pointer or IP)

⇓

instruction to be run next

⇒ stack pointer & associated
frame pointer

⇓

Used to manage the stack for
fn parameters, local variables
& return addresses.

Process API

- Create
- Destroy
- Wait
- Misc control ⇒ like suspend process
- Status

Process creation

Load program into memory from disk

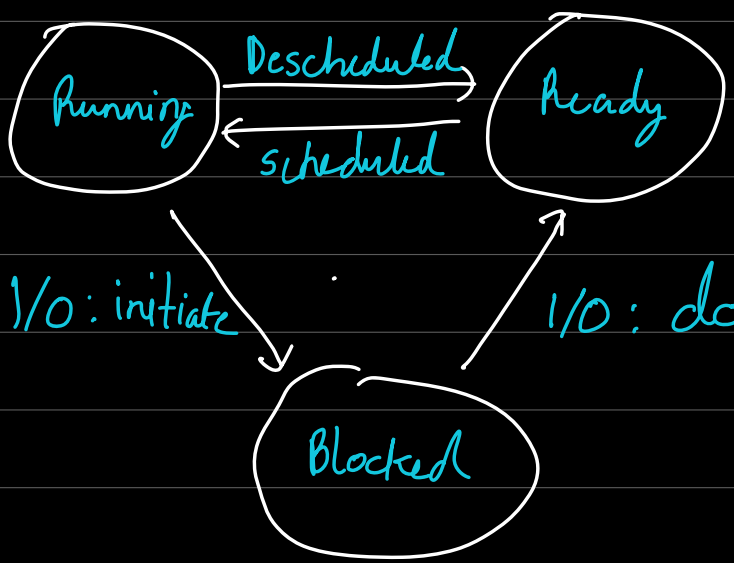
- Eagerly - old OSs
- Lazily - modern OSs
  - * Paging
  - * Swapping

} more of this in later chapters

Unix each process by default 3 open file descriptors ⇒ standard input, output & error

Process States

- Running
- Blocked
- Ready



Process State transitions

Data structures

OS needs to track a bunch of things

related to process.

- process list ⇒ ready, running processes etc.
- blocked processes, when I/O done, schedule them to run

structure is called Process control block (PCB) or process descriptor

- In Linux, seems like "task_struct" in "sched.h" is that data structure. Code on Github.