Uniprocessor Scheduling

Co-operative Multi-tasking: When processes co-ordinate who gets to run.
Pre-emptive Multi-tasking: When processor co-ordinates which processes will run.

Scheduling Types: 4 main types of scheduling.

- **Long-term:**
  - Decides which processes run at all.
  - Controls transition from *new* to *ready to run* state. (where it falls into domain of *Medium, short-term* schedulers)
  - Aims to estimate and schedule to run a balance of I/O limited and CPU limited processes.
- **Medium-term:**
  - Involves swapping processes to/from disk.
  - Bringing back a swapped process puts it under *Short-term* scheduler's domain
- **Short-term:**
  - Called *Dispatcher.* Deals with what happens *now.*
  - Runs *very* frequently.
  - Aims to run process that maximize CPU utilization, and I/O device utilization concurrently
  - Runs when:
    - Process Yields
    - Process Terminates
    - Process Blocked due to:
      - I/O Operation
      - Semaphore/Mutex
      - Page-faults
      - Interrupt Handling
      - Many other *System calls...*
- **I/O based (later):**

Rate Limiting Step: Processes' *turn-around time* limited due to:

- **I/O Operations:** waiting for I/O to respond
- **CPU Operations:** using CPU extensively in *bursts*
  - CPUs have imporved at a faster rate than I/O - most processes tend to be I/O limited

Scheduling Criteria:

- **Turnaround Time:** Total time taken to complete a process (includes I/O, scheduling delays and CPU usage time)
- **Response Time:** Time b/w being issued a command vs. responding to it.
  - *e.g.* Windows search tool shows results as it searches to minimize response time.
  - Not as important for *Daemons* (Background Processes)
- **Deadlines:** Real-time systems need turn-around times based on strict *deadlines*
  - *e.g.* a blu-ray player needs to read data, decrypt, decode, and display in within a set time-frame to prevent video quality degredation.
- **Predictability:** The turn-around time of a process should be consistent. Inconsistency leaves schedulers and users confused.
- **Throughput:** A system should aim to maximize number of processes completed in unit time.
- **Processor Utilization:** Maximum efforts should be maid to not leave processor free.
- **Fairness:** Although a system should have *priority*-based scheduling, it should be fairly imposed so that no one is left *starving.*
- **Priortization:** Priorties (stored in integers) are assigned to each process. Higher-priority process

runs more frequently.

- **Balancing Resources:** Schedule processes in a manner that keeps *both* the CPU and the I/O busy at the same time.