scheduling

- remember that we have various types of queues: a ready queue with threads that are ready to go, a I/O wait queue, a network wait queue, memory wait queue etc. after things are done reading/writing data from these, they get put back on the ready queue and then will need to get scheduled.
- There are lots of different metrics one could optimize for while scheduling
  - average latency of a job
  - throughput (num jobs completed per unit time)
  - shortest time spent waiting by jobs
  - etc
- accordingly, we can choose different algorithms for scheduling
  - shortest time first jobs (if we can estimate job length) → this will starve long jobs
  - first come first served
  - round robin
  - lottery scheduling: different jobs get a different number of lottery tickets. schedule a random job with likelihood of a job being scheduled proportional to number of lottery tickets it holds.
- remember that if you switch a lot, context switching is a huge cost, but also cache thrashing is a huge huge cost.
- how to evaluate a scheduling algorithm?
  - deterministic modeling – takes a predetermined workload and compute the performance of each algorithm for that workload
  - queuing models – mathematical approach for handling stochastic workloads
  - implement + simulate: – build system which allows actual algorithms to be run against actual data. most flexible/general.
- if there are some very long jobs and some short ones and you do FIFO, then the short jobs could really get screwed → in terms of time to completion from submitting metric
- on the other hand, if you have a lot of the same length jobs, if you do round robin, each of them will take a very long time to complete (other than having lots of context switches and cache thrashes) → in terms of time to completion from submitting metric
- it can be shown that if response time is the metric (time to completion from submitting job), you can prove that shortest to completion first is provably optimal → think about why
- but note that shortest time to completion algorithm can starve long jobs!