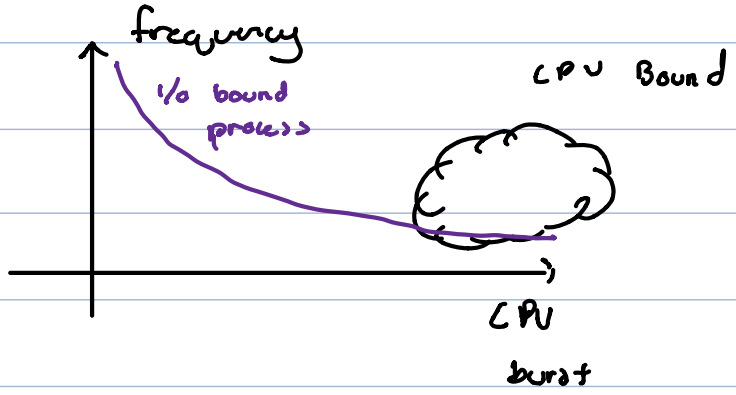
CPU Scheduling

Assigning Process to the CPU

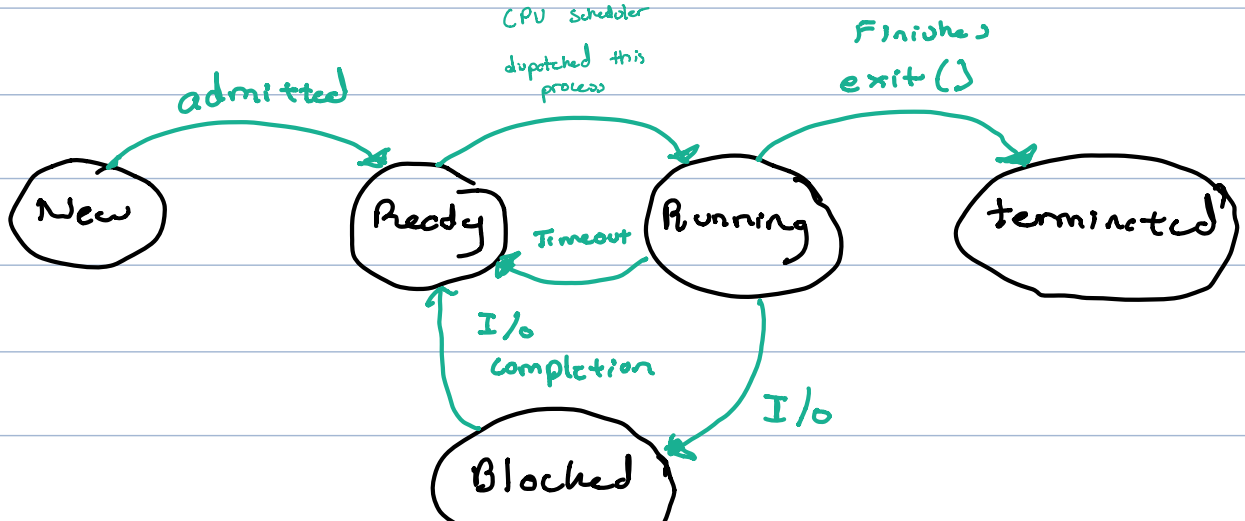
Short term scheduling

1. Nature of Process
   - ↳ CPU burst
   - ↳ I/O



2. When are scheduling decisions made?
   - a. A new process arrives
   - b. A process blocked
   - c. A process terminates
   - d. A process goes from blocked to ready
   - e. A process goes from running to ready

c, b → Another process must be served



Non- preemptive

A process runs until completion or it blocks

Preemptive

A running process is interrupted and placed in the ready queue

How to evaluate different schedulers?

1. CPU utilization
2. Turn-arround time [waiting in the queue + job complete]
3. Response time / responsiveness
4. Throughpot (How many processes per unit time the system is able to get out)
5. Waiting time [waiting in the queue]
6. Fairness / priority / deadline.

1. FCFS (non-preemptive)
   - First come first served give CPU served

   give CPU to the first requesting process

   Advantages: Simple

   Disadvantages: Average waiting time

Ex

Doesn't have to wait

P_1	20
P_2	2
P_3	3

avg waiting time

had to wait for 20s

$$\frac{0 + 20 + 22}{3} = 14$$

P_2	2
P_3	3
P_1	20

$$\frac{0 + 2 + 5}{3} = 2.3$$ average waiting time

(2) Shortest Job first (Shortest process next)

Non-preemptive scheduler

Associate with every process, the length of it's next CPU burst

Assign the CPU to the process with the shortest CPU burst

STF is optimal minimum average waiting time.

$$T_{n+1} = \alpha\, t_n + (1 - \alpha)\, T_n \qquad 0 \le \alpha \le 1$$

T_{n+1}: estimate

t_n: Last burst size (Last value)

T_n: History

Exponential weighted moving average

$\alpha = 0.5$

$T_1 = 5$

$t_1 = 6$

$\alpha * t_n + (1 - \alpha) T_n$

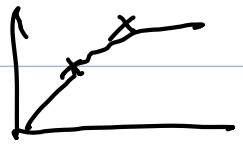
$T_2 = 0.5 \times 6 + 0.5 \cdot 5$

$= 5.5$

$t_2 = 8$

$T_3 = 0.5 * 8 + 0.5 \cdot 5.5$

$= 4 + 2.75 = 6.75$



Disadvantage:

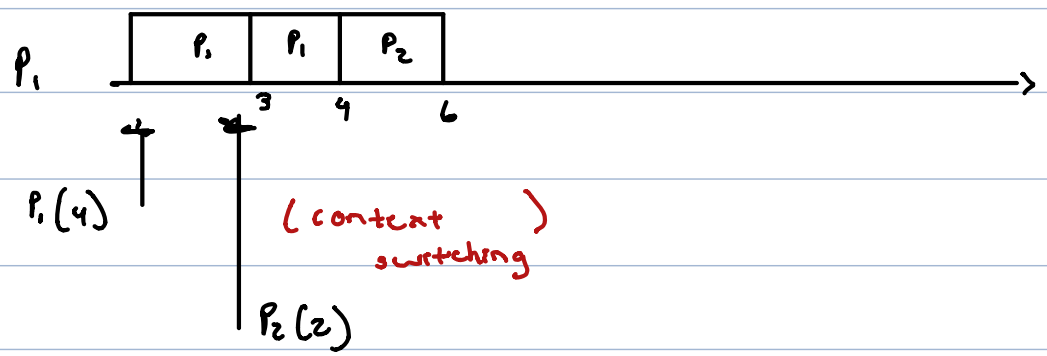
Shortest job first may lead to starvation

[3] Shortest Remaining Time first

Preemptive

- Lower turn-around time than STF

(shorter jobs are given immediate attention



[4] Highest Response ratio next

Non preemptive

Minize the normalized turnaround time

Response Ratio $\rightarrow$ $R = \frac{w + s}{s}$

w: time spent waiting

s: expected service time

Pick process with highest R.

⋮

5 Round Robin

Preemptive (based on a clock)

Interrupts are generated every time slice (quantum)

10 - 100 msec

Short quatum => Overhead context switching

Long quatum => Approaches FCFS