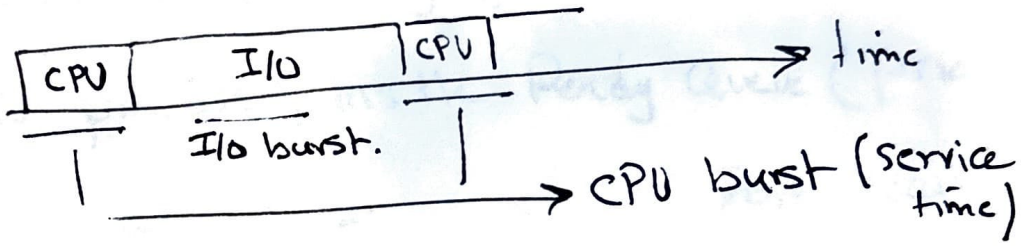
Scheduling

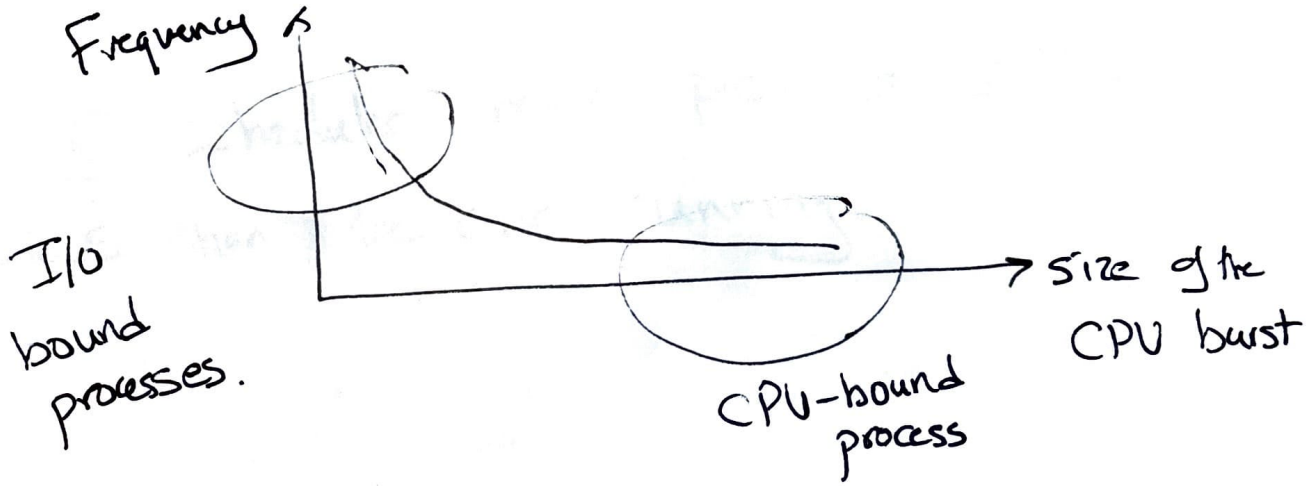
Assigning a process to be executed on the CPU.

Short-term Scheduling

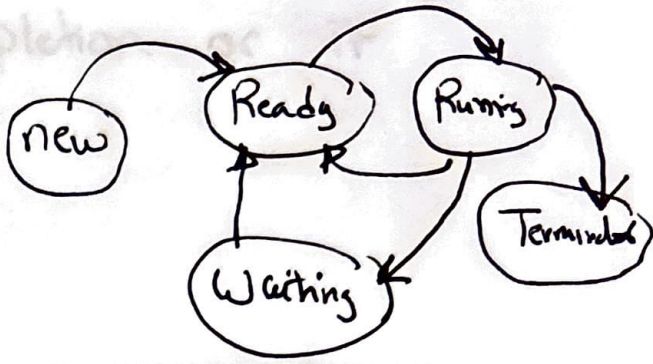
I Nature of Processes

Process:





[2] When are Scheduling decisions made? [2]



[a] When a process completes

[b] when a process goes to Waiting

[c] when a Running process goes to Ready (Time out).

[d] when a ~~New~~ process joins the Ready Queue (priority).

[e] " " " " from Waiting

[a], [b] scheduler must pick a different process than the one running.

Non preemptive

A process must run to Completion or it blocks itself (e.g I/O).

Preemptive

A running process may be interrupted and placed in back in the Ready queue.

[a] Timeout.
[b] A process moves from waiting to Ready. } priority
[c] A new process arrive to Ready queue }

[Work-Conserving Schedules] ⟹ Server can only go idle if no process is in the Ready queue.

Non-Work Conserving Schedules

How to evaluate/Compare between Schedulers? [4]

[1] Utilization of the CPU
[2] Turnaround time / Waiting Time. in the Ready Queue. / Response Time
[3] Throughput.
[4] Fairness.
[5] Power Consumption
[6] Overhead in deciding
[7] load balancing.

---

[1] FCFS First Come First Serve.

Non-preemptive.

Adv

So Simple.

Dis Adv

Long average waiting Time.

	Service time
P_1	20
P_2	2
P_3	3

P_1

$$\text{avg. waiting Time} = \frac{0 + 20 + 22}{3} = 14.$$

P_2
P_3
P_1

avg. Waiting Time $= \frac{0+2+5}{3} = 7/3 = 2.3$

---

[2] Shortest Job First (SJF)
Shortest Process Next (SPN).

Non-preemptive.

Assign for every process an Eshimate of the CPU burst required.

assign the CPU to the process with shortest CPU burst.

P_1	6
P_2	2
P_3	4
P_4	1

$\Longrightarrow$ P_4 P_2 P_3 P_1

Optimal: [gives the minimum average waiting time for a set of processes] ~~that exists~~

P_4 P_2 P_3 P_1

avg. waiting time $= \frac{0 + 1 + 3 + 7}{4} = \frac{11}{4}$.

Hard to know the ~~CPU~~ next CPU burst of a process.

EWMA: Exponentially Weighted Moving Average

$$E_{n+1} = \alpha M_n + (1-\alpha) E_n$$

latest — history

M_n ↓ last measurement

E_n ↓ previous estimate

	CPU burst M
n=1	2
n=2	4
n=3	6
.	4
.	8
.	.
.	.
(n+1) →	→ prediction. Estimate

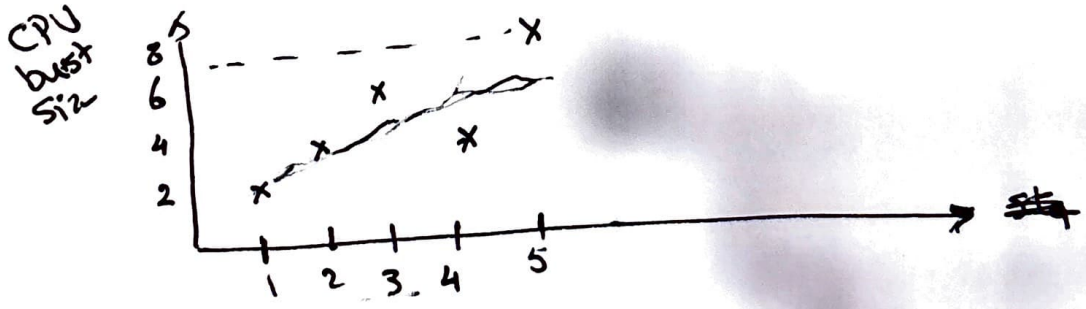
$0 \leq \alpha \leq 1$ weight.

$\alpha = 0.5$ $\quad E_2 = 2$

$$E_3 = \overset{\alpha}{0.5} * \overset{M_2}{\boxed{4}} + \overset{\alpha}{0.5} * \overset{E_2}{\boxed{2}} =$$

$$2 + 1 = \boxed{3}$$

$$E_4 = 0.5 * 6 + 0.5 * 3 = 4.5$$



$$E_{n+1} = \alpha M_n + (1-\alpha) E_n$$

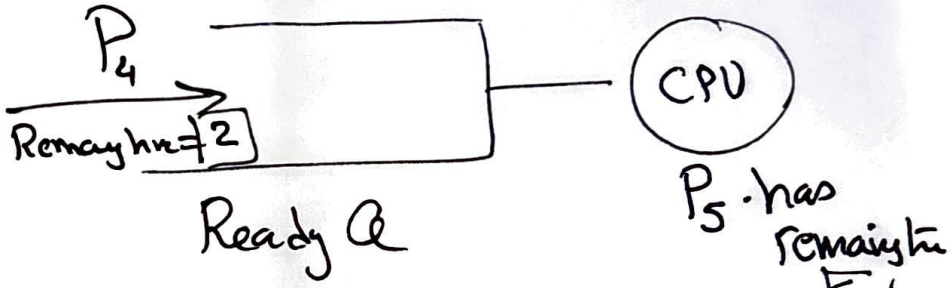
$$= \alpha M_n + (1-\alpha)\left[\alpha M_{n-1} + (1-\alpha) E_{n-1}\right]$$

↓

---

3 Shortest Remaing Time First (SRTF)

Preemptive.



Preempt. (P_4 will run).

Better alg. Turnaround time than SJF because shorter process are given immediate attention.