CPU SCHEDULING ALGORITHMS WITH EXAMPLES

🎾 **First Come First Serve(FCFS) - Nonpreemptive**

🎾 **Shortest Job First(SJF) - Nonpreemptive**

🎾 **Shortest Remaining Time First(SRTF) - Preemptive SJF**

🎾 **Nonpreemptive Priority Scheduling**

🎾 **Preemptive Priority Scheduling**

🎾 **Round Robin Scheduling(RR) - Preemptive**

🥎 **In case of priority scheduling and SJF, preemptive or nonpreemptive can be mentioned indirectly**

🥎 **"scheduling takes place when the running process gets blocked on IO or running process finishes it's cpu burst"**

🎾 **SJF is also called as shortest remaining processing time first(SRPT) or SRT**

TYPE OF QUESTIONS ON CPU SCHEDULING ALGORITHMS

🎾 **Arrival time is given as 0 or equal for all process**

   🎾 **If given the order explicitly, follow that**

   🎾 **If not given, assume the order given in table**

🎾 **Problem with different arrival times**

🎾 **Problem with IO burst time along with CPU burst time**

🎾 **Problem with context switch overhead**

$\rightarrow P_1 \quad 0$

$\rightarrow P_2 \quad 0$

$\rightarrow P_3 \quad 0$

SCHEDULING CRITERIA PARAMETERS

🎾 **Turnaround time = completion time - arrival time**

🎾 **Waiting time = turnaround time - total cpu burst - total io burst**

🎾 **Response time = start time - arrival time**

🎾 **CPU utilization = (total cpu burst / max (completion time))*100**

🎾 **Throughput = total process / (max(completion time) -min(arrival time))**

TAT = CT - AT
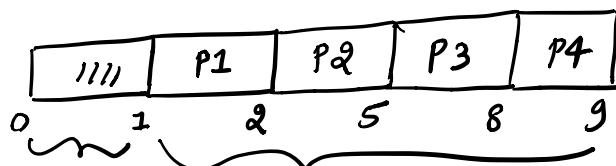**WT = TAT - CPU BT - IO BT**
RT = ST - AT

FCFS SCHEDULING ALGORITHM

🎾 **If you have multiple process in ready queue, schedule the one with least arrival time among them**

	AT	BT	ST	CT	TAT	WT	RT
P1	1	1	1	2	1	0	0
P2	2	3	2	5	3	0	0
P3	2	3	5	8	6	3	3
P4	3	1	8	9	6	5	5

Ready: P1 P2 P3 P4

TAT = CT - AT
WT = TAT - CPU BT - IO BT
RT = ST - AT

////	P1	P2	P3	P4

0   1   2   5   8   9

$$\text{CPU UT} = \frac{8}{9} \times 100$$

$$\text{Avg TAT} = \frac{(1+3+6+6)}{4}$$

$$\text{Avg WT} = (5+3)/4$$

$$\text{Avg RT} = (3+5)/4$$
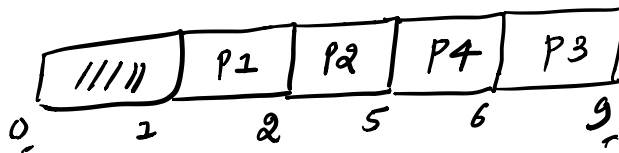
$$\text{Throughput} = \frac{4}{9-01} = \frac{4}{8} = \frac{1}{2}$$

SJF SCHEDULING ALGORITHM

🎾 **If you have multiple process in ready queue, schedule the one with <u>least burst time</u> among them**

	AT	BT	ST	CT	TAT	WT	RT
P1	1	1	1	2	1	0	0
P2	2	→3	2	5	3	0	0
P3	2 1	→3	6	9	7	4	4
P4	3	1	5	6	3	2	2

Ready : P̶1̶ P̶2̶ P3 P̶4̶

```
 ┌──────┬────┬────┬────┬────┐
 │ ///))│ P1 │ P2 │ P4 │ P3 │
 └──────┴────┴────┴────┴────┘
 0      1    2    5    6    9
```

TAT = CT - AT
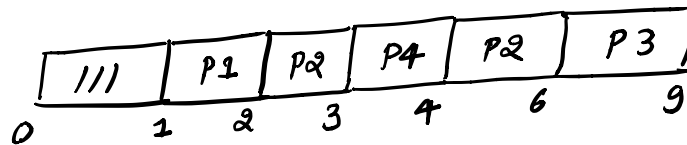WT = TAT - CPU BT - IO/BT
RT = ST - AT

SRTF SCHEDULING ALGORITHM

🎾 **Preemptive SJF - At each unit of time, check ready queue for any new process.**

	AT	BT	ST	CT	TAT	WT	RT
P1	1	1 ←	1	2	1	0	0
P2	→ 2	3 2	2	6	4	1	0
P3	2	3	6	9	7	4	4
P4	→ 3	1	3	4	1	0	0

Ready : P1 P2 P3 P4

TAT = CT - AT
WT = TAT - CPU BT - IO BT
RT = ST - AT

///	P1	P2	P4	P2	P3
0	1	2	3	4	6	9

NONPREEMPTIVE PRIORITY SCHEDULING

🎾 **If you have multiple process in ready queue, schedule the one with highest priority among them**

	AT	BT	Priority	ST	CT	TAT	WT	RT
P1	1	1	4 (L)	1	2	1	0	0
P2	2	3	- 3	6	9	7	4	4
P3	2	3	- 2 ←	2	5	3	0	0
P4	3	1	1 (H)	5	6	3	2	2

Ready : P1 P2 P3 P4

////	P1	P3	P4	P2
0	1	2	5	6	9
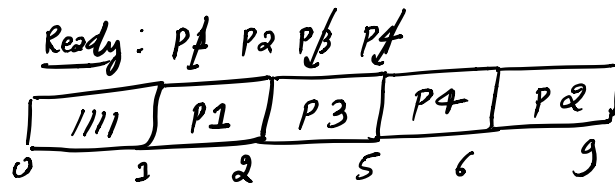
TAT = CT - AT
WT = TAT - CPU BT - IO BT
RT = ST - AT

PREEMPTIVE PRIORITY SCHEDULING

🎾 **Also check for priority of new processes arriving in ready queue**

	AT	BT	Priority	ST	CT	TAT	WT	RT
P1	1	1	4 (L)	1	2	1	0	0
P2	2	3	3	6	9	7	4	4
P3	2	3/2	2	2	6	4	1	0
P4	3	1	1 (H)	3	4	1	0	0

TAT = CT - AT
WT = TAT - CPU BT - IO/BT
RT = ST - AT

Ready : P1 P2 P3 P4

//))	P1	P3	P4	P3	P2

0   1   2   3   4   6   9
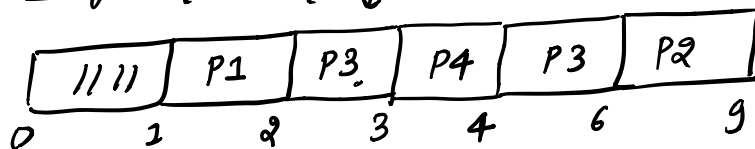
ROUND ROBIN SCHEDULING ALGORITHM

🎾 **Preemptive FCFS**

🎾 **Preempt each process after timeslice/quantum expires**

	AT	BT	ST	CT	TAT	WT	RT
P1	1	1 0	1	2	1	0	0
P2	2	3 2 1	2	8	6	3	0
P3	2	3 2 1	3	9	7	4	1
P4	3	1 0	4	5	2	1	1

Quantum : __1 ms__

Ready : P1 P2 P3 P4 P2 P3 P2 P3

TAT = CT - AT
WT = TAT - CPU BT - IO BT
RT = ST - AT