Process scheduling algorithm

- **"Process scheduling** is the set of polices and mechanisms supported by OS, that control the order in which the work to be done is completed".
- The primary object of scheduling to optimize system performance of the system because it determines which processes will wait and which will progress.
- A Process Scheduler schedules different processes to be assigned to the CPU based on particular scheduling algorithms.
- These algorithms are either **non-preemptive** or **preemptive**.
    i) **Non-preemptive algorithms** are designed so that once a process enters the running state, it cannot be preempted until it completes its allotted time
    ii) **The preemptive scheduling** is based on priority where a scheduler may preempt a low priority running process anytime when a high priority process enters into a ready state.

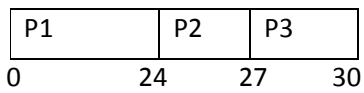
- There are **six popular process scheduling algorithms** which we are going to discuss:
    1. First-Come, First-Served (FCFS) Scheduling
    2. Shortest-Job-Next (SJN) Scheduling
    3. Priority Scheduling
    4. Shortest Remaining Time
    5. Round Robin(RR) Scheduling
    6. Multiple-Level Queues Scheduling

1. First Come, First Served (FCFS):

- ✓ Jobs are executed on first come, first served basis.
- ✓ It is a **non-preemptive** scheduling algorithm.
- ✓ Easy to understand and implement.
- ✓ Its implementation is based on FIFO queue.
- ✓ Poor in performance, as average wait time is high.

Process	Burst Time	waiting time	turnaround time
P1	24 (min/ms)	00	24
P2	3 (min/ms)	24	27
P3	3 (min/ms)	27	30

GANTT chart using FCFS:



Average waiting time = $\frac{0+24+27}{3} = \frac{51}{3}$ = **17min**

Av. turnaround time = $\frac{24+27+30}{3} = \frac{81}{3}$ = **27min**

Throughput = $\frac{3}{30}$ = **0.1 min**

2. Shortest Job Next (SJN)

- ✓ This is also known as **shortest job first**, or **SJF**.
- ✓ This is a **non-preemptive scheduling** OR **Preemptive scheduling** algorithm.
- ✓ Best approach to **minimize waiting time**.
- ✓ Shortest time to completion first.
- ✓ Impossible to implement in interactive systems where the required CPU time is not known.
- ✓ The processer should know in advance how much time a process will take.

Process	Arrival time (AT)	Burst Time (BT)	Completion time(CT)	TAT=CT-AT	W.T.=TAT-BT (non preemptive)
P1	0.0	07 (min/ms)	07	07-00 = 07	(07-07) = 00
P2	2.0	04 (min/ms)	12	12-02 = 10	(10-04) = 06
P3	4.0	01 (min/ms)	08	08-04 = 04	(04-01) = 03
P4	5.0	04 (min/ms)	16	16-05 = 11	(11-04) = 07

GANTT chart Non- preemptive scheduling:

P1	P3	P2	P4
0	7	8	12	16

Average waiting time = $\frac{0+6+3+7}{4} = \frac{16}{4}$ = **4min**

A.TAT = 8min

3. Shortest Remaining Time

- ✓ **Shortest remaining time** (SRT) is the preemptive version of the SJN algorithm.
- ✓ The processor is allocated to the job closest to completion but it can be preempted by a newer ready job with shorter time to completion.
- ✓ Impossible to implement in interactive systems where required CPU time is not known.
- ✓ It is often used in batch environments where short jobs need to be given preference.

Process	Arrival time (AT)	Burst Time (BT)	Completion time(CT)		
P1	0	7 /5	16	16	9
P2	2	4 /2	7	5	1
P3	4	1 done	5	1	0
P4	5	4	11	6	2

GANTT chart Preemptive scheduling:

P1	P2	P3	P2	P4	P1
0	2	4	5	7	11	16

Average waiting time = $\frac{9+1+0+2}{4} = \frac{12}{4}$ = **3min**

A.TAT = 7min

4. Priority based Scheduling:

- ✓ Priority scheduling is a **non-preemptive algorithm** and one of the most common scheduling algorithms in **batch systems**.
- ✓ Each process is assigned a priority.
- ✓ Process with **highest priority is to be executed first** and so on.
- ✓ Processes with **same priority** are executed on **first come first served (FCFS)** basis.
- ✓ Priority can be decided based on memory requirements, time requirements or any other resource requirement.

Process	Burst Time (BT)	Priority	TAT	W.T
P1	10	3	16	6
P2	1	1	1	0
P3	2	4	18	16
P4	1	5	19	18
P5	5	2	6	1

GANTT chart Priority Based scheduling:

P2	P5	P1	P3	P4

0 1 6 16 18 19

Average waiting time = $\frac{6+0+16+18+1}{5} = \frac{41}{5}$ **= 8.2min**

5. Round Robin Scheduling:

- ✓ Round Robin is a **preemptive** process scheduling algorithm.
- ✓ Each process is provided a fix time to execute; it is called a **quantum or time slice**.
- ✓ Once a process is executed for a given time period, it is preempted and other. Process executes for a given time period.
- ✓ **Context switching** is used to save states of preempted processes.

TQ= 4

Process	Burst Time (BT)	Priority	TAT	W.T
P1	23	3	29	(10-4)=6
P2	3	1	7	4
P3	3	4	10	7

GANTT chart Priority Based scheduling:

P1	P2	P3	P1	P1	P1	P1	P1

0 4 7 10 14 18 22 26 29

Average waiting time = 6+4+7 / 3 = 5.6

Process	Burst Time (BT)		TAT	W.T
P1	53	51-20= 31 31-20=13	134	(121-40)=81
P2	17	0	37	20
P3	68	68-20=48 48-20=28 28-20=8	162	134-40=94
P4	24	24-20=4	121	117-20=97

TQ = 20

GANTT chart Priority Based scheduling:

0 20 37 57 77 97 117 121 134 154 162

6. Multiple - Level Queues Scheduling:

- ✓ Multiple-level queues are not an independent scheduling algorithm.
- ✓ They make use of other existing algorithms to group and schedule jobs with common characteristics.
  - ➔ Multiple queues are maintained for processes with common characteristics.
  - ➔ Each queue can have its own scheduling algorithms.
  - ➔ Priorities are assigned to each queue.
  - ➔ In foreground working with round –robin and background working with FCFS.
- ✓ For example, CPU-bound jobs can be scheduled in one queue and all I/O-bound jobs in another queue. The Process Scheduler then alternately selects jobs from each queue and assigns them to the CPU based on the algorithm assigned to the queue