* Scheduling Algorithms:

1) First Come First Served (FCFS):

A simple approach to maintain a queue, that is, the item that comes first will be served first. The ready queue can also be maintained with this approach.

The arriving process is added onto the tail of the queue & the process at the head of the queue is dispatched to the processor for execution. In this way, a ready queue is implemented as a linked list wherein the order of arrival of the processes is maintained with the simple logic of first in, first out (FIFO).

This scheduling policy is non-pre-emptive because the process which has arrived first will be executed first to its completion.

For eg:

Process	CPU Burst time (ms)
P1	24
P2	3
P3	3

Gantt chart:

Waiting time (WT)	Turn around Time (TAT)
0	24
24	27
27	30

Average WT = 17ms   Average TAT = 27ms

Case 2: When P1, P2 & P3 arrives in order

P2 → P3 → P1

Gantt chart

P2	P3	P1
0	3	6	30
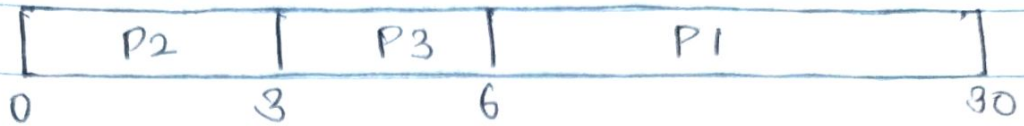
	Waiting time	Turn around time
P1	6	30
P2	0	3
P3	3	6
Average	3ms	13ms

Note that in Case 1, AWT = 17ms whereas in case 2, AWT = 3ms only. So, case-2 is much better than case-1. This is because in case-1, the small processes (P2 & P2) had to wait longer for bigger processes to release the CPU. This effect is known as Convoy Effect wherein all the other processes end up waiting for one big process to get the CPU. This effect results in lower CPU & device ultilization than might be possible if the shorter process were allowed to go first.
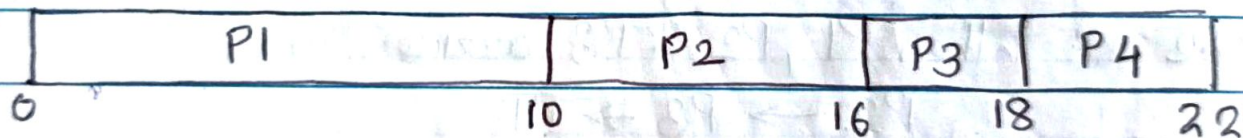
Advantages:
1. It is a simple algorithm & easy to implement.
2. It is suitable for Batch systems.

→ Disadvantages:
1. The average waiting time is not minimal.
2. It is not suitable for time sharing systems
3. A proper mix of CPU & I/o bound jobs is required achieve good results from FCFS.

eg:

Process	Arrival time	Burst Time
PD	0	10
P1	1	6
P2	3	2
P3	5	4

P1		P2	P3	P4
0	10	16	18	22

Process	Arrival time (t_0)	Burst time (Δt)	Finish time (t_1)	TAT $=(T_1 - t_0)$	WT $= TAT - \Delta t$
P1	0	10	10	10	0
P2	1	6	16	15	9
P3	3	2	18	15	13
P4	5	4	22	17	13