Scheduling Algorithms

A Process Scheduler schedules different processes to be assigned to the CPU based on particular scheduling algorithms. There are six popular process scheduling algorithms which we are going to discuss in this chapter −

A. First-Come, First-Served (FCFS) Scheduling
B. Shortest-Job-Next (SJN) Scheduling
C. Priority Scheduling
D. Shortest Remaining Time
E. Round Robin(RR) Scheduling
F. Multiple-Level Queues Scheduling

These algorithms are either **non-preemptive or preemptive**. Non-preemptive algorithms are designed so that once a process enters the running state, it cannot be preempted until it completes its allotted time, whereas 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.

First Come First Serve (FCFS)

1. Jobs are executed on first come, first serve basis.
2. It is a non-preemptive, pre-emptive scheduling algorithm.
3. Easy to understand and implement.
4. Its implementation is based on FIFO queue.
5. Poor in performance as average wait time is high.

Wait time of each process is as follows −

Process	Wait Time : Service Time - Arrival Time
P0	0 - 0 = 0
P1	5 - 1 = 4
P2	8 - 2 = 6
P3	16 - 3 = 13

Average Wait Time: (0+4+6+13) / 4 = 5.75

Problem 1

Consider the given table below and find Completion time (CT), Turn-around time (TAT), Waiting time (WT), Response time (RT), Average Turn-around time and Average Waiting time.

Process ID	Arrival time	Burst time
P1	2	2
P2	5	6
P3	0	4
P4	0	7
P5	7	4

Solution

For this problem CT, TAT, WT, RT is shown in the given table –

Process ID	Arrival time	Burst time	CT	TAT=CT-AT	WT=TAT-BT	RT
P1	2	2	13	13-2= 11	11-2= 9	9
P2	5	6	19	19-5= 14	14-6= 8	8
P3	0	4	4	4-0= 4	4-4= 0	0
P4	0	7	11	11-0= 11	11-7= 4	4
P5	7	4	23	23-7= 16	16-4= 12	12

Average Waiting time = (9+8+0+4+12)/5 = 33/5 = 6.6 time unit (time unit can be considered as milliseconds)

Average Turn-around time = (11+14+4+11+16)/5 = 56/5 = 11.2 time unit (time unit can be considered as milliseconds)

Shortest Job Next (SJN)

1. This is also known as **shortest job first**, or SJF
2. This is a non-preemptive, pre-emptive scheduling algorithm.
3. Best approach to minimize waiting time.
4. Easy to implement in Batch systems where required CPU time is known in advance.

5. Impossible to implement in interactive systems where required CPU time is not known.
6. The processer should know in advance how much time process will take.

Given: Table of processes, and their Arrival time, Execution time

Process	Arrival Time	Execution Time	Service Time
P0	0	5	0
P1	1	3	5
P2	2	8	14
P3	3	6	8

Waiting time of each process is as follows −

Process	Waiting Time
P0	0 - 0 = 0
P1	5 - 1 = 4
P2	14 - 2 = 12
P3	8 - 3 = 5

Average Wait Time: (0 + 4 + 12 + 5)/4 = 21 / 4 = 5.25

Priority Based Scheduling

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

Given: Table of processes, and their Arrival time, Execution time, and priority. Here we are considering 1 is the lowest priority.

Process	Arrival Time	Execution Time	Priority	Service Time
P0	0	5	1	0
P1	1	3	2	11
P2	2	8	1	14
P3	3	6	3	5

Waiting time of each process is as follows −

Process	Waiting Time
P0	0 - 0 = 0
P1	11 - 1 = 10
P2	14 - 2 = 12
P3	5 - 3 = 2

Average Wait Time: (0 + 10 + 12 + 2)/4 = 24 / 4 = 6

Shortest Remaining Time

1. Shortest remaining time (SRT) is the preemptive version of the SJN algorithm.
2. 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.
3. Impossible to implement in interactive systems where required CPU time is not known.
4. It is often used in batch environments where short jobs need to give preference.

Round Robin Scheduling

1. Round Robin is the preemptive process scheduling algorithm.
2. Each process is provided a fix time to execute, it is called a **quantum**.
3. Once a process is executed for a given time period, it is preempted and other process executes for a given time period.
4. Context switching is used to save states of preempted processes.

Wait time of each process is as follows −

Process Wait Time : Service Time - Arrival Time

P0 (0 - 0) + (12 - 3) = 9

P1 (3 - 1) = 2

P2 (6 - 2) + (14 - 9) + (20 - 17) = 12

P3 (9 - 3) + (17 - 12) = 11

Average Wait Time: (9+2+12+11) / 4 = 8.5

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.

1. Multiple queues are maintained for processes with common characteristics.
2. Each queue can have its own scheduling algorithms.
3. Priorities are assigned to each queue.

Multilevel Queue Scheduling

 Ready queue is partitioned into separate queues:
foreground (interactive)
background (batch)
 Each queue has its own scheduling algorithm
o foreground – RR
o background – FCFS Scheduling must be done between the queues
o Fixed priority scheduling; (i.e., serve all from foreground then from background). Possibility of starvation.
o Time slice – each queue gets a certain amount of CPU time which it can schedule amongst its processes; i.e., 80% to foreground in RR
o 20% to background in FCFS

Multilevel Feedback Queue Scheduling

 A process can move between the various queues; aging can be implemented this way
 Multilevel-feedback-queue scheduler defined by the following parameters:
o number of queues
o scheduling algorithms for each queue

o method used to determine when to upgrade a process
o method used to determine when to demote a process
o method used to determine which queue a process will enter when that process needs service