- **Algorithms used:**

  - **Round Robin Scheduling Algorithm:**

1: Create an array remainbt[] to keep track of remaining burst time of the processes and should have an initial copy of bursttime[].

2: Create an array waitingtime[] that will store waiting times of processes and should be initially initialized as 0.

3: Initial time assigned to time, t should be zero i.e., t=0.

4: Traverse all the processes until all is completed and follow i process if its not done.

if remainbt[]>quantum

i. t = t + quantum;
ii. remainbt[i] - = quantum;

else

i. t = t + bursttime_remain[i];
ii. waitingtime[i] = t – bursttime[i];
iii. remainbt[i] = 0;

And hence the process gets over.

  - **Multilevel Queue Algorithm:**

1: As process starts the execution, it initially enters queue 1.

2: If the process in queue 1 executes for a fixed unit and completes in that fixed unit or it gives system for I/O operation in that fixed unit, that doesn’t change the priority. If it again comes in ready queue, then it needs to start execution again in queue 1.

3: If process of queue 1 is not possible in that fixed unit then the priority of that process reduces and is then shifted to queue 2.

4: The last 2 points are also valid for queue 2 but having the time quantum double of that fixed unit.

If the process fails to execute in that fixed time quantum then it is shifted to the lower priority queue.

5: Processes are scheduled in First Come First Serve (FCFS) manner in the last queue.

6: Whenever the higher priority queues are empty, that’s when a process in lower priority can execute.

7: Process arriving in the higher priority queue interrupts the process running in the lower priority queue.