Name: Kaniz Fatema
ID: 20245103154
Intake: 53
Section: 01

Comment on your findings and write down the challanges you faced to implement those problems.

FCFS (First Come First Serve)

While solving the problem for FCFS, I first used arrays to store the data, but I got wrong output. Then I checked the logic multiple times and realized that using arrays made it difficult to manage multiple attributes like arrival time, burst time together.

Then I tried using a vector of structs and solved the problem. It became easier to sort the processes based on arrival time and store all the data in one place.

Next, I noticed that the equation I was using for waiting time and turnaround time was not giving the correct result. So, I switched to the equations taught in theory class.

Turnaround Time = Completion Time – Arrival Time

Waiting Time = Turnaround Time – Burst Time

Using these formulas gave me the correct output.

SJF Non-Preemptive

This algorithm was harder than FCFS because I had to select the shortest job from the set of arrived processes, and select the shortest burst time among the arrived processes also handle CPU idle time if no process had arrived at the time. As a result, The loop was tricky to implement

After some trial and error, I was able to correctly set the idle time and execute the shortest available job. I used a vector and sorted it based on arrival time and burst time.

SJF Preemptive (SRTF)

I found SJF Preemptive the most difficult to implement. I had to simulate every single time unit, and at each step. I had to check which process should run based on the shortest remaining time.

I couldn't solve it just by sorting once like in non-preemptive. I had to use the ready queue concept . Every time step, I checked all arrived and unfinished processes, and selected the one with the smallest remaining time. If no process had arrived yet, I had to make sure the CPU remains idle, which took a few tries to get right.

Finally, after storing and updating remaining_time separately and handling the edge cases, I got the correct output.