Report on CPU Scheduler

## BY

## Revati Goyal

## Enrollment No. 21113129

## Year : 4th

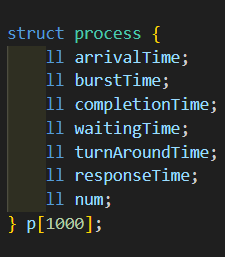
CPU Scheduler

CPU scheduling is a process which allows one process to use the CPU while the execution of another process is on hold (in waiting state) due to unavailability of any resource like I/O etc, thereby making full use of CPU. The aim of CPU scheduling is to make the system efficient, fast and fair.

I made 4 CPU Scheduling Algorithm

1. First Come First Serve
2. Shortest Job First
3. Shortest Remaining Job First
4. Round Robin

For building algorithm I used a struct data structure

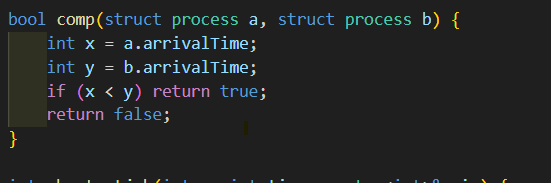


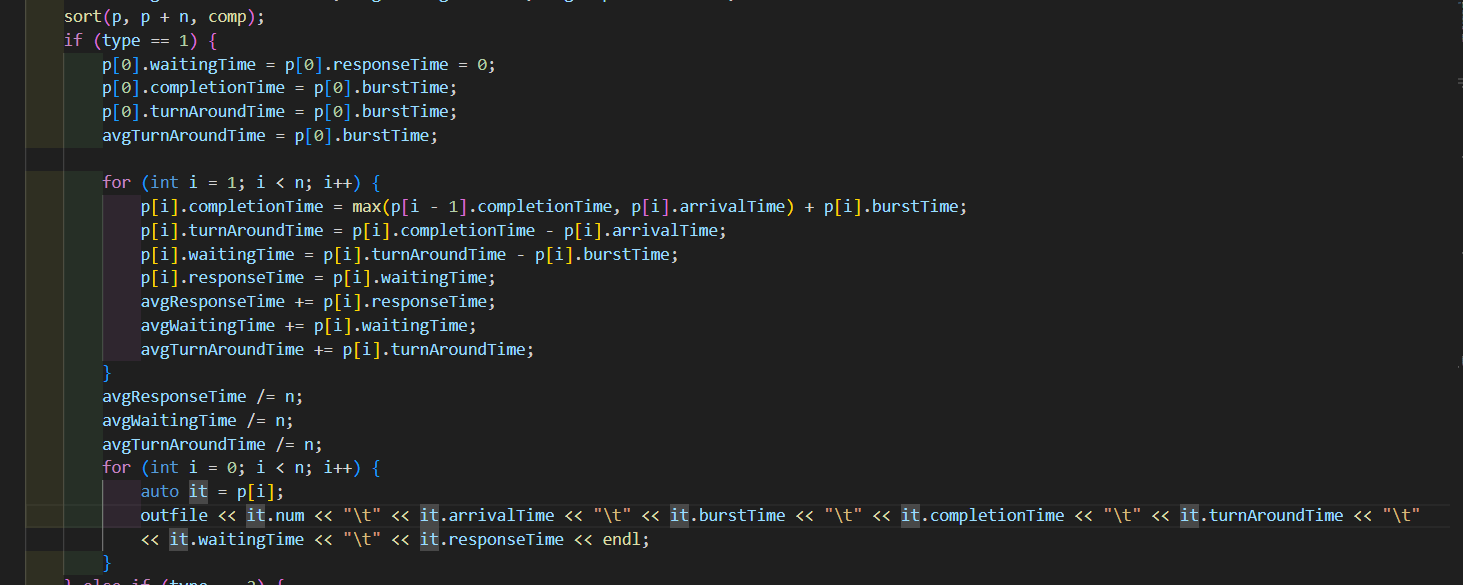
It has all details about the process the serial number,arrival time and burst time that would be input by the User and will store calculated completion time,waiting time, turn around time and response time for the type of scheduler used

First the user is required to input the type of scheduler user want to use

And the number of processes to be executed. After which inputs of arrival time and burst time for each process is to be entered.

Then I sort the processes on the basis of their arrival time with the help of comp( function



**First Come First Serve**

It’s Non Preemptive Algorithms thus waiting time= response time.

In FCFS first arrived process is executed completely first, since I have sorted the processes on the basis of arrival time it is much easier to calculate.

Completion time depends on two process first arrival time of the current process and the completion time of the previous process after which only current process can start.

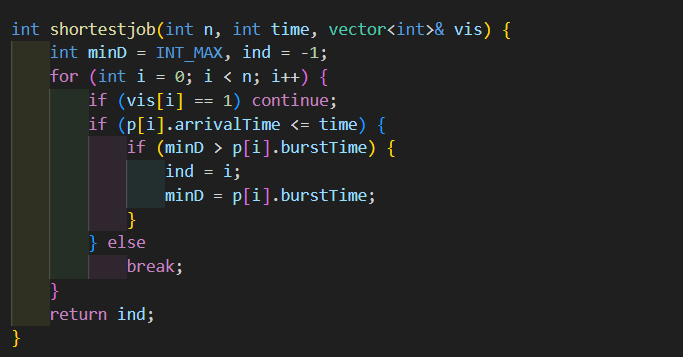
Turn Around Time = Completion Time – Arrival Time

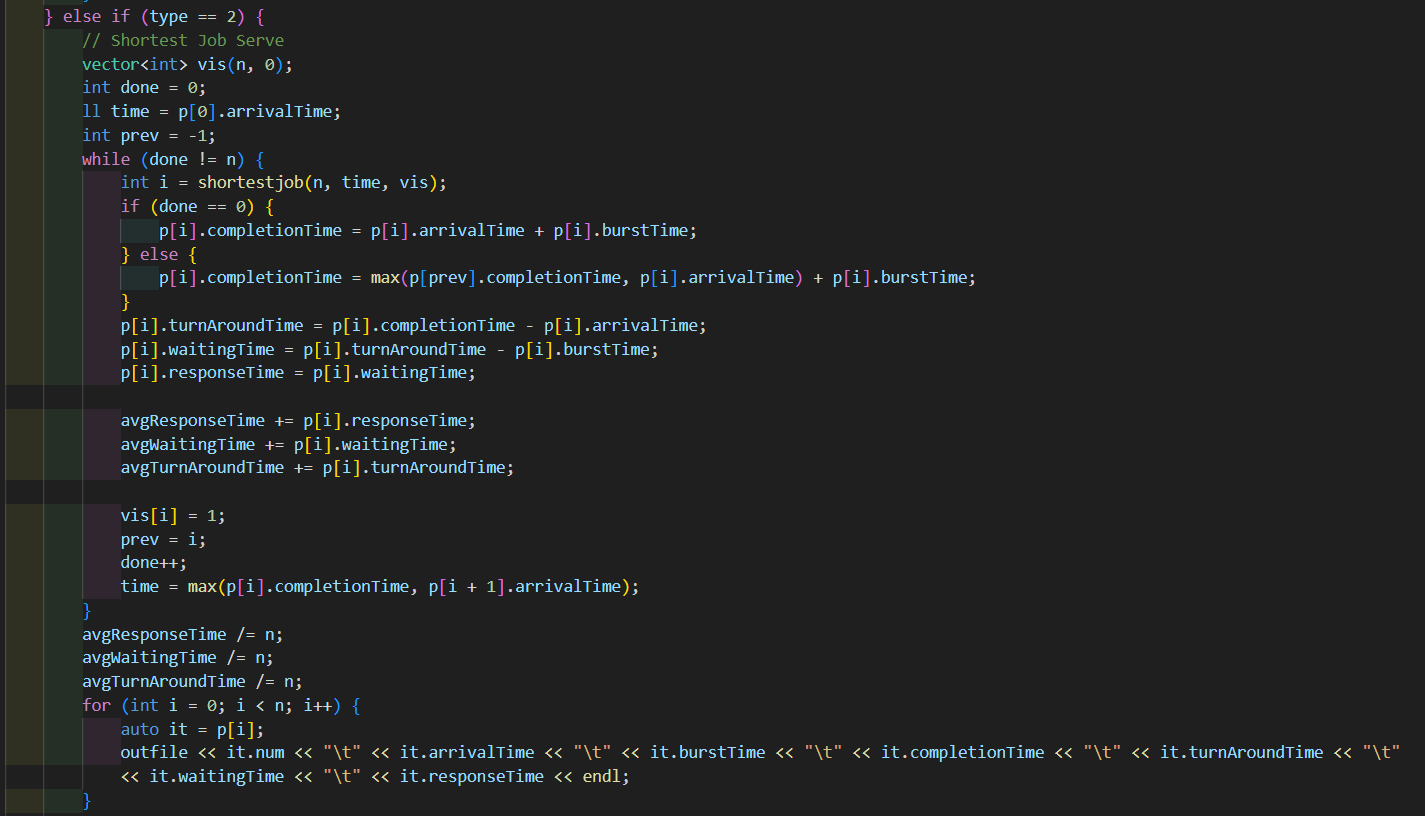
Waiting Time = TAT – Burst Time

Response Time = Time at which process started – Arrival time

**Shortest Job First**

It is also a Non Preemptive Scheduling Algorithm thus process which execute completely.

It selects the processes with same arrival time on the basis of their burst time.

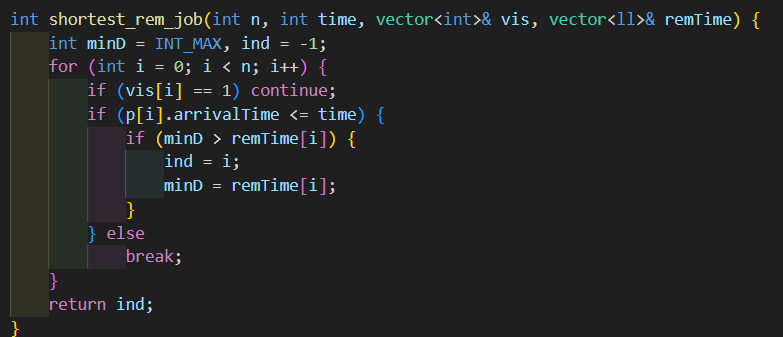
This function gives me the process with shortest burst time and not executed yet among all the processes having arrival time lesser than the current time. 

I used a while loop which will continue until all the processes are executed. It gives me the index of the of shortest job. Then it’s completion time, waiting time, and reponse time is calculated.

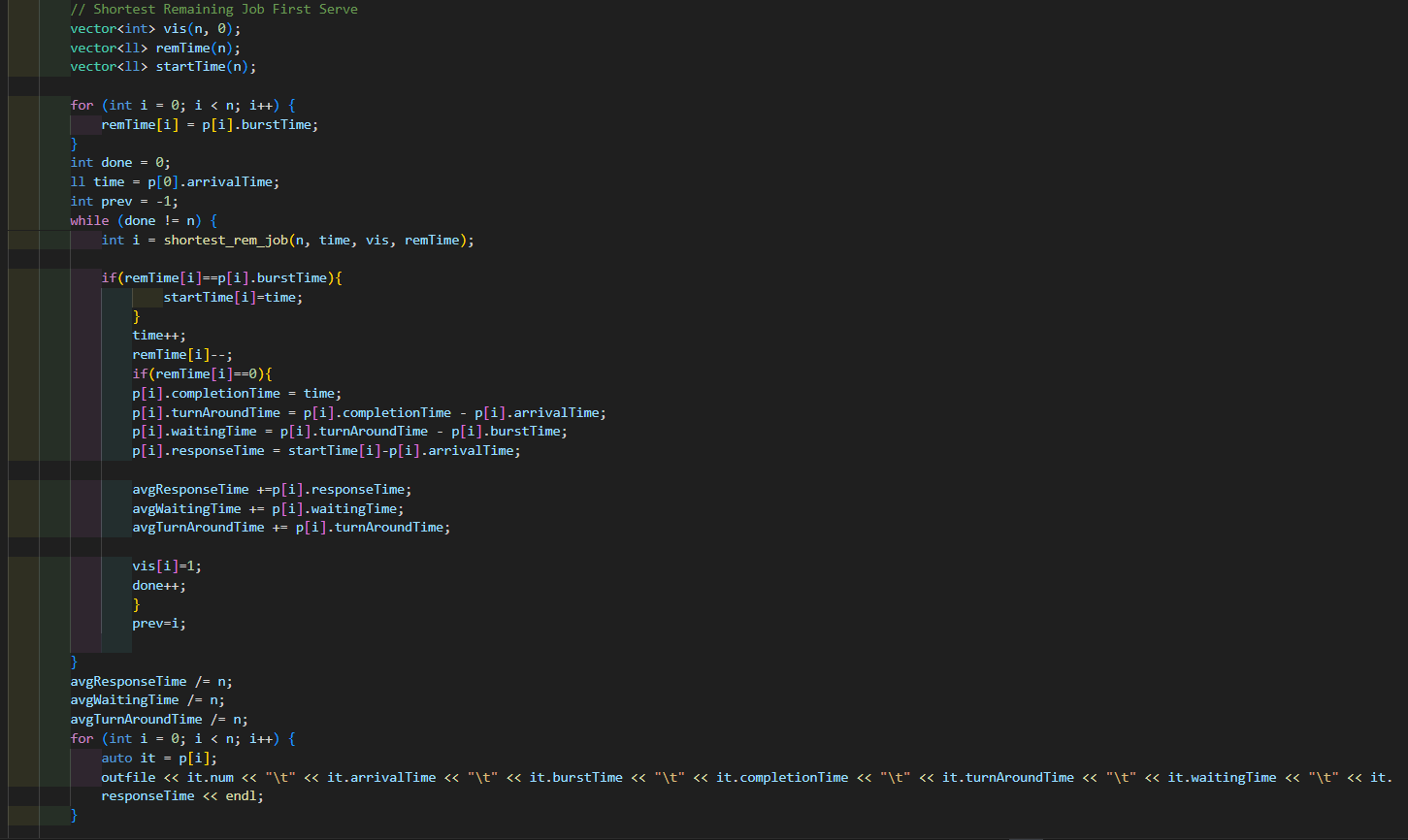
After which the job is marked as visited to avoid extra execution.

**Shortest Remaining Time First**

It’s a preemptive process that executes on the basis of smallest remaining time.



Here I used a variable named remTime that tells me the remaing time of process. This will be the basis of comparison for this algorithm unlike burst time for Shortest Job First.

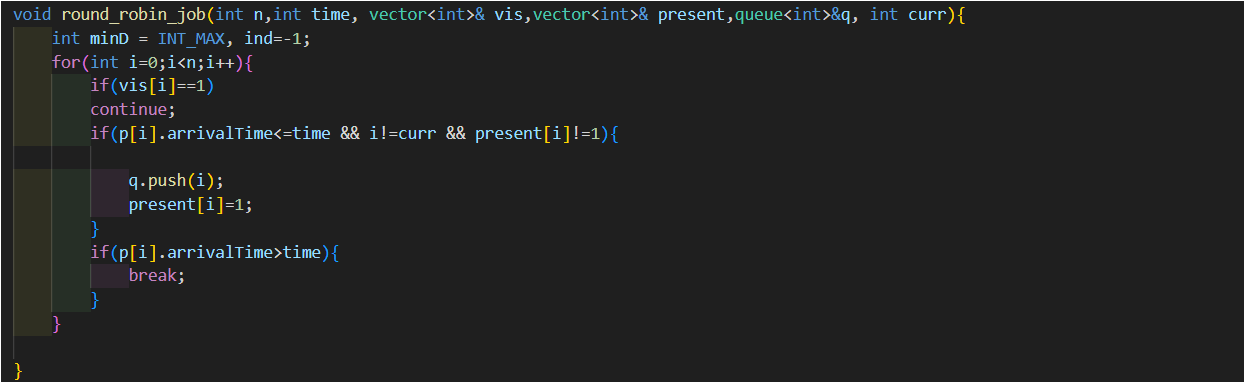


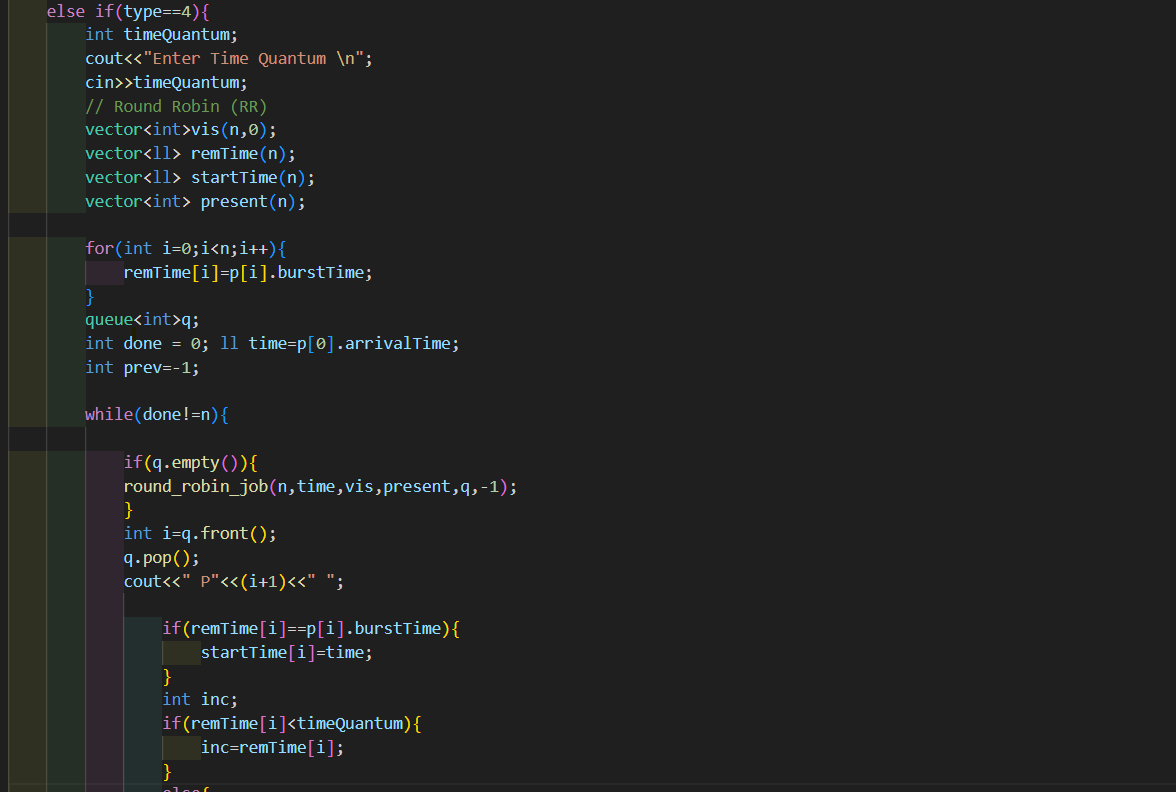
In this algorithm response time will be different from waiting time. Thus I need to keep track of start time for each algorithm and then when remtime is 0 it calculates all the values.

**Round Robin**

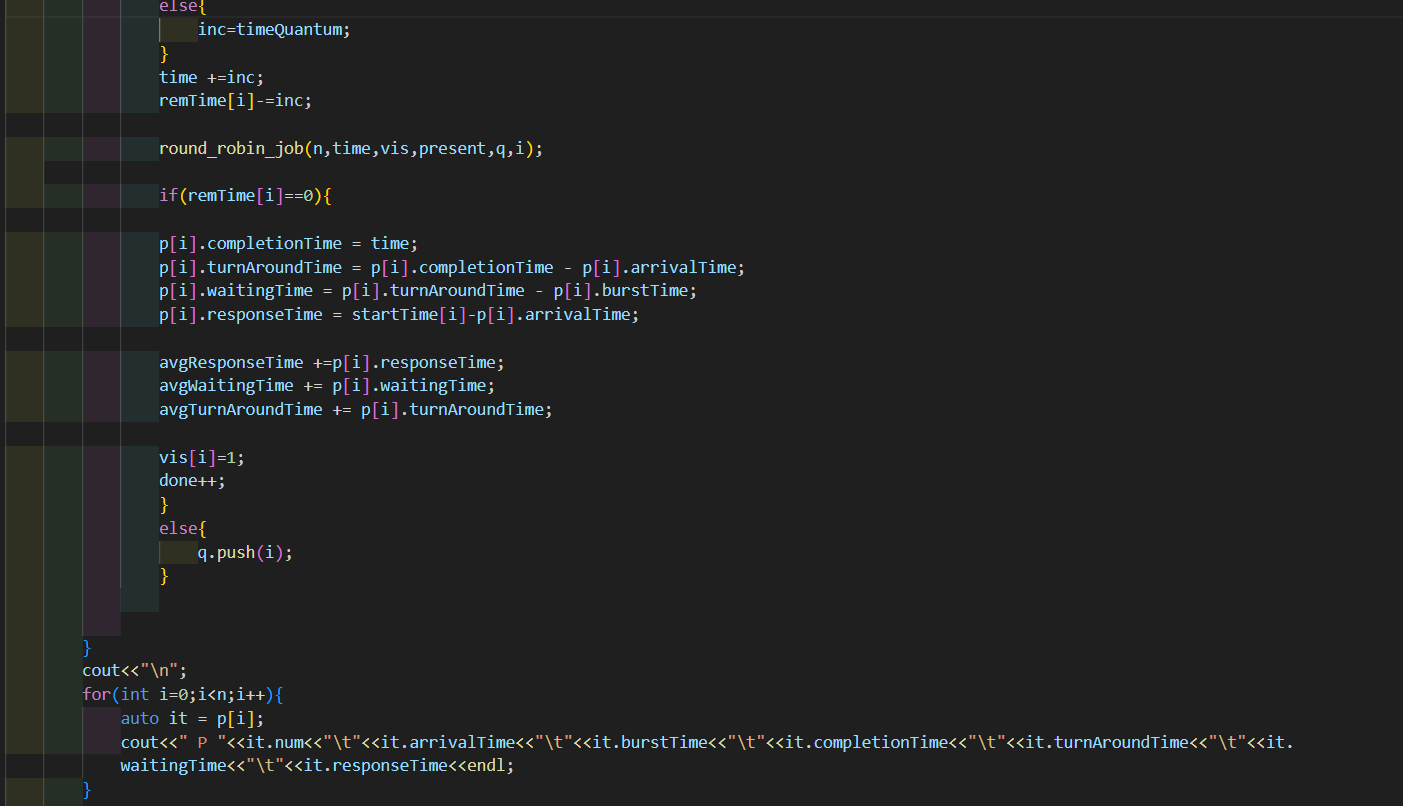
I used a ready queue to keep processes which are in waiting.

Thus all the processes which have arrive will be pushed to the ready queue except process that is currently running.

****

It takes time quantum as input and then if none of the process is in the queue it runs round robin function. After that it executes max of time quantum , remTime of the process. ****

If the process is completed it’s parameters are calculated. And it will be marked visited. But before that while execution of the process some other algorithms could have also arrived that are taken into account by round\_robin() funct. If the process is not finished it is pushed into the queue and the loop continues until all the processes are completely executed.

****