

NAME :- Shaikh Areeba Mohammed Ismail
ROLL NO.:- 46
CLASS/BATCH:- TE-B-2

Practical No:- 04

// Write a program to simulate CPU Scheduling Algorithms: FCFS, SJF
(Preemptive), Priority (Non-
Preemptive) and Round Robin (Preemptive).

SJF (Preemptive) CODE

```
import java.util.Scanner; class Process {  
    int pid; // Process ID int at; // Arrival Time int bt; // Burst Time  
    int ct; // Completion Time int tat; // Turnaround Time int wt;  
    // Waiting Time  
    boolean done; // To mark if process is completed  
}  
  
public class SJF {  
    public static void main(String[] args) { Scanner sc = new  
        Scanner(System.in);  
        System.out.print("Enter number of processes: "); int n = sc.nextInt();  
  
        Process[] p = new Process[n];  
  
        for (int i = 0; i < n; i++) { p[i] = new Process(); p[i].pid = i + 1;  
            System.out.print("Enter Arrival Time of P" + (i + 1) + ": "); p[i].at = sc.nextInt();  
            System.out.print("Enter Burst Time of P" + (i + 1) + ": "); p[i].bt = sc.nextInt();  
        }  
        int completed = 0, currentTime = 0; float totalTAT = 0, totalWT = 0;  
  
        // SJF scheduling (non-preemptive) while (completed < n) {  
        int idx = -1;  
        int minBT = Integer.MAX_VALUE;  
  
        // Find process with minimum BT among arrived & not done for (int i = 0; i < n; i++) {  
        if (!p[i].done && p[i].at <= currentTime) { if (p[i].bt < minBT) {  
            minBT = p[i].bt; idx = i;  
        }  
        }  
        }  
  
        if (idx == -1) {  
            // No process arrived yet, increment time currentTime++;  
        } else {  
            // Execute selected process p[idx].ct = currentTime + p[idx].bt;  
            p[idx].tat = p[idx].ct - p[idx].at;
```

```

p[idx].wt = p[idx].tat - p[idx].bt; p[idx].done = true;

currentTime = p[idx].ct; completed++;

totalTAT += p[idx].tat; totalWT += p[idx].wt;
}
}

// Print Table System.out.println("\nPID\tAT\tBT\tCT\tTAT\tWT");
for (int i = 0; i < n; i++) {
System.out.println("P" + p[i].pid + "\t" + p[i].at + "\t" + p[i].bt +
"\t" +
p[i].ct + "\t" + p[i].tat + "\t" + p[i].wt);
}

System.out.println("\nAverage Turnaround Time = " + (totalTAT / n));
System.out.println("Average Waiting Time = " + (totalWT / n));
}
}

```

OUTPUT :

```

gescoe@gescoe-OptiPlex-3010:~/Desktop/TE_B[46]/LP-1$ javac SJF.java
gescoe@gescoe-OptiPlex-3010:~/Desktop/TE_B[46]/LP-1$ java SJF Enter
number of processes: 3
Enter Arrival Time of P1: 0
Enter Burst Time of P1: 5 0
Enter Arrival Time of P2:
Enter Burst Time of P2: 3 0
Enter Arrival Time of P3:
Enter Burst Time of P3: 8
PID    AT     BT      CT      TAT      WT
P1      0      5       8       8       3
P2      0      3       3       3       0
P3      0      8      16      16      8

Average Turnaround Time = 9.0 Average Waiting Time = 3.6666667
gescoe@gescoe-OptiPlex-3010:~/Desktop/TE_B[46]/LP-1$
```