



**CET-225**  
**Operating Systems**  
**Experiment # 08**

**Experiment Title**

CPU Scheduling Algorithm-FCFS ,SJF

**Assessment of CLO(s): 04**

**Performed on** \_\_\_\_\_

<b>Student Name:</b>			
<b>Roll No.</b>		<b>Group</b>	
<b>Semester</b>		<b>Session</b>	

<b>Total (Max)</b>	<b>Performance (03)</b>	<b>Viva (03)</b>	<b>File (04)</b>	<b>Total (10)</b>
<b>Marks Obtained</b>				
<b>Remarks (if any)</b>				

**Experiment evaluated by**

<b>Instructor's Name</b>	<b>Engr. Bushra Aziz</b>		
<b>Date</b>		<b>Signature</b>	

## Lab Experiment 8

### FCFS CPU SCHEDULING ALGORITHM

**First Come First Serve** is a Non-preemptive Scheduling algorithm where each process is executed according to its arrival time.

**Step 1:** Input the number of processes required to be scheduled using FCFS, burst time for each process and its arrival time.

**Step 2:** Using enhanced bubble sort technique, sort the all given processes in ascending order according to arrival time in a ready queue.

**Step 3:** Calculate the Finish Time, Turnaround Time and Waiting Time for each process which in turn help to calculate Average Waiting Time and Average Turnaround Time required by CPU to schedule given set of process using FCFS.

**Step 4:** Process with less arrival time comes first and gets scheduled first by the CPU.

**Step 5:** Calculate the Average Waiting Time and Average Turn Around Time.

**Step 6:** Stop.

#### Sample Run:

Enter total number of processes (maximum 20):3

Enter Process Arrival Time and Burst time.

Calculate Waiting time and turnaround time for each process

Process	Arrival time	Burst time
P1	0 ms	18 ms
P2	2 ms	7 ms
P3	2 ms	10 ms

**Gantt Chart**

P1		P2		P3	
0 ms	18 ms	18 ms	25 ms	25 ms	35 ms

Calculate Waiting time and turnaround time for each process

Process	Waiting Time	Turnaround Time
P1	0ms	18ms
P2	16ms	23ms
P3	23ms	33ms

Total waiting time:  $(0 + 16 + 23) = 39\text{ms}$

Average waiting time:  $(39/3) = 13\text{ms}$

## Lab Experiment 8

Total turnaround time:  $(18 + 23 + 33) = 74\text{ms}$

Average turnaround time:  $(74/3) = 24.66\text{ms}$

### SJF CPU SCHEDULING ALGORITHM

**Shortest job first (SJF)** or shortest job next, is a scheduling policy that selects the waiting process with the smallest execution time to execute next. SJN is a non-preemptive algorithm.

**Step 1:** Input the number of processes required to be scheduled using SJF, burst time for each process and its arrival time.

**Step 2:** Using selection sort technique, sort the all given processes in ascending order according to burst time in ascending order.

**Step 3:** Calculate the Finish Time, Turnaround Time and Waiting Time for each process which in turn help to calculate Average Waiting Time and Average Turnaround Time required by CPU to schedule given set of process

**Step 4:** Process with less Burst and arrival time comes first and gets scheduled first by the CPU.

**Step 5:** Calculate the Average Waiting Time and Average Turn Around Time.

**Step 6:** Stop.

#### Sample Run:

Enter number of process: 4

Enter Process Arrival Time and Burst time.

Process	Arrival time	Burst time
P1	3 ms	5 ms
P2	0 ms	4 ms
P3	4 ms	2 ms
P4	5 ms	4 ms

**Gantt Chart**

P2		P3		P4		P1	
0ms	4ms	4ms	6ms	6ms	10ms	10ms	15ms

Calculate Waiting time and turnaround time for each process

Process	Waiting Time	Turnaround Time
P1	7ms	12ms
P2	0ms	4ms
P3	0ms	2ms
P4	1ms	5ms

## Lab Experiment 8

Total waiting time:  $(7 + 0 + 0 + 1) = 8\text{ms}$

Average waiting time:  $(8/4) = 2\text{ms}$

Total turnaround time:  $(12 + 4 + 2 + 5) = 23\text{ms}$

Average turnaround time:  $(23/4) = 5.75\text{ms}$

### Terms and formulas used in above scheduling algorithms:

**Completion Time:** Time at which process completes its execution.

**Turn Around Time:** Time Difference between completion time and arrival time.

**Turn Around Time** = Completion Time – Arrival Time

**Waiting Time (W.T):** Time Difference between turnaround time and burst time.

**Waiting Time** = Turn Around Time – Burst Time.

## **Lab Experiment 8**

### **Exercises**

1. Write a Python program to implement and simulate the FCFS Algorithm.
2. Write a Python program to implement and simulate the SJF Algorithm.
3. Modify both algorithms for the different arrival time.