

# Operating Systems

## Lab 04: CPU Scheduling Algorithms (FCFS and SJF)

### Goals:

In this lab, you will simulate the following CPU scheduling algorithms:

- **First Come First Served (FCFS):** In FCFS scheduling, each process will be CPU scheduled based on its arrival time in the ready queue.
- **Non-preemptive Shortest Job First (SJF):** The SJF scheduling algorithm scheduled processes for running on CPU based on their CPU burst time; the process with the shortest CPU burst time is scheduled first. If two processes with the same CPU burst time exists in the ready queue, then FCFS scheduling approach is applied.

*Given  $n$  processes with their burst times, the task is to find average waiting time and average turnaround time using FCFS and SJF scheduling algorithms.*

### Policies:

It should go without saying that all the work that you will turn in for this lab will be yours. Do not surf the web to get inspiration for this assignment, do not include code that was not written by you. You should try your best to debug your code on your own, but it's fine to get help from a colleague as long as that means getting assistance to identify the problem and doesn't go as far as receiving source code to fix it (in writing or orally).

### Contents:

Write a C program that simulates the FCFS and non-preemptive SJF CPU scheduling algorithms. Repeat this experiment with different sets of processes and trace the waiting and turnaround times for each process.

Here we are considering that arrival time for all processes is **0**.

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

**Turnaround Time:** Time Difference between completion time and arrival time.

$$\text{Turnaround Time} = \text{Completion Time} - \text{Arrival Time}$$

**Waiting Time:** Time Difference between turnaround time and burst time.

$$\text{Waiting Time} = \text{Turn Around Time} - \text{Burst Time}$$

In this task, we have assumed arrival times as 0, so turn around and completion times are same.

- 1- Input the processes along with their burst time (bt).
- 2- Find waiting time (wt) for all processes.
- 3- As first process that comes need not to wait so waiting time for process 1 will be 0 i.e.  $wt[0] = 0$ .
- 4- Find **waiting time** for all other processes i.e. for process i ->  
 $wt[i] = bt[i-1] + wt[i-1]$  .
- 5- Find **turnaround time** = waiting\_time + burst\_time for all processes.
- 6- Find **average waiting time** =  
 $total\_waiting\_time / no\_of\_processes$ .
- 7- Similarly, find **average turnaround time** =  
 $total\_turn\_around\_time / no\_of\_processes$ .

Expected Results:

FCFC:

```
Enter total number of processes(maximum 20):3
Enter Process Burst Time
P[1]:24
P[2]:3
P[3]:3

Process      Burst Time      Waiting Time      Turnaround Time
P[1]          24              0                24
P[2]          3              24              27
P[3]          3              27              30

Average Waiting Time:17
Average Turnaround Time:27
```

SJF:

```
Enter number of process:4
Enter Burst Time:
p1:4
p2:8
p3:3
p4:7

Process      Burst Time      Waiting Time      Turnaround Time
p3           3              0                3
p1           4              3                7
p4           7              7              14
p2           8              14             22

Average Waiting Time=6.000000
Average Turnaround Time=11.500000
```

**What should you submit:**

- 1) Please submit a zip file which includes the C code of the two algorithms:**
- 2) Your zip file should be named as: “姓名\_lab4”**
- 3) Please submit your zip file by sending email to me as attachment:  
my email address is: hzb564@jnu.edu.cn**