

**Experiment-4**

Operating Systems

**Program:**

MCA

**SVKM'S-NMIMS**

**Mukesh Patel School of Technology Management & Engineering**

**School of Technology Management and Engineering**

**[2024-25]**

**Lab Manual**

PART B

| Roll No.: A073 | Name: Aryan Srivastava |
| --- | --- |
| Class: MCA FY | Batch: B3 |
| Experiment Number- 4 | |
| Date of Experiment: | Date of Submission: 20 aug |
| Grade: |  |

**B.1 Program with Output to be written by student**

# time dependence jo pahele aaya voh pahele execute

# after the first proicess then see the shortest burst time

#aryan srivastava A073 B3

def sort\_by\_burst\_time(processes):

    for i in range(len(processes)):

        for j in range(i + 1, len(processes)):

            if processes[i]['burst\_time'] > processes[j]['burst\_time']:

                processes[i], processes[j] = processes[j], processes[i]

def calculate\_times(processes):

    total\_burst\_time = 0

    total\_waiting\_time = 0

    total\_turnaround\_time = 0

    print("Process\tBurst Time\tWaiting Time\tTurnaround Time")

    for process in processes:

        burst\_time = process['burst\_time']

        waiting\_time = total\_burst\_time

        turnaround\_time = waiting\_time + burst\_time

        print(f"{process['name']}\t{burst\_time}\t\t{waiting\_time}\t\t{turnaround\_time}")

        total\_burst\_time += burst\_time

        total\_waiting\_time += waiting\_time

        total\_turnaround\_time += turnaround\_time

    avg\_waiting\_time = total\_waiting\_time / len(processes)

    avg\_turnaround\_time = total\_turnaround\_time / len(processes)

    print(f"\nAverage Waiting Time: {avg\_waiting\_time:.2f}")

    print(f"Average Turnaround Time: {avg\_turnaround\_time:.2f}")

processes = [

    {'name': 'P1', 'burst\_time': 6},

    {'name': 'P2', 'burst\_time': 8},

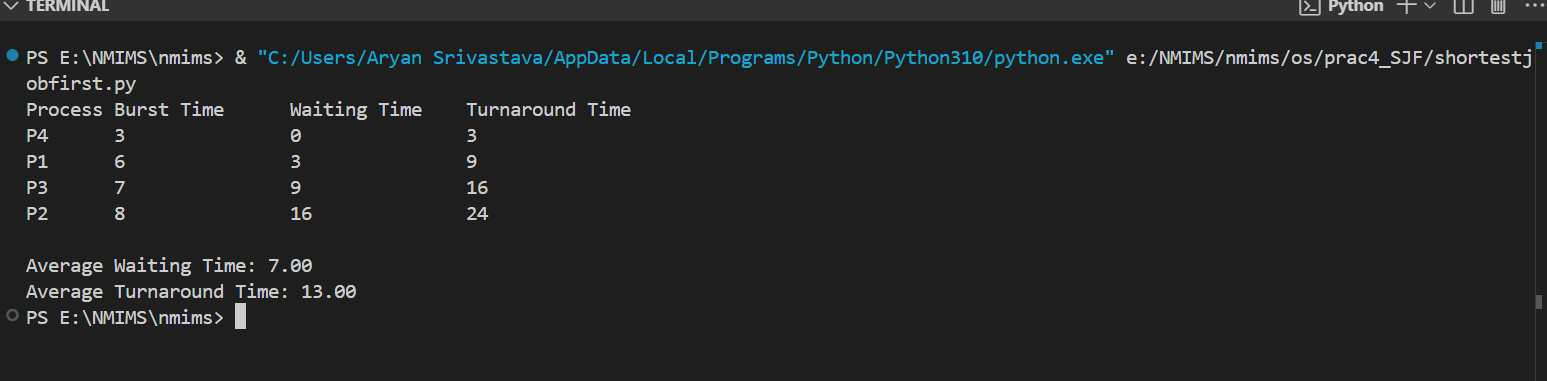
    {'name': 'P3', 'burst\_time': 7},

    {'name': 'P4', 'burst\_time': 3}

]

sort\_by\_burst\_time(processes)

calculate\_times(processes)

OUTPUT

**B.2 Answers the following question based on study**

1. Distinguish between Preemptive and Nonpreemptive CPU scheduling?  
     
   Preemptive Scheduling allows the operating system to interrupt a running process to assign the CPU to another process with higher priority. It's more responsive but can cause more context switching.

nonpreemptive Scheduling ensures that a running process keeps the CPU until it either completes or voluntarily yields it. This reduces context switching but may lead to lower system responsiveness.