

**Experiment-3**

Operating Systems

**Program:**

MCA

**SVKM'S-NMIMS**

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

**School of Technology Management and Engineering**

**[2024-25]**

Lab Manual

**Experiment No.03**

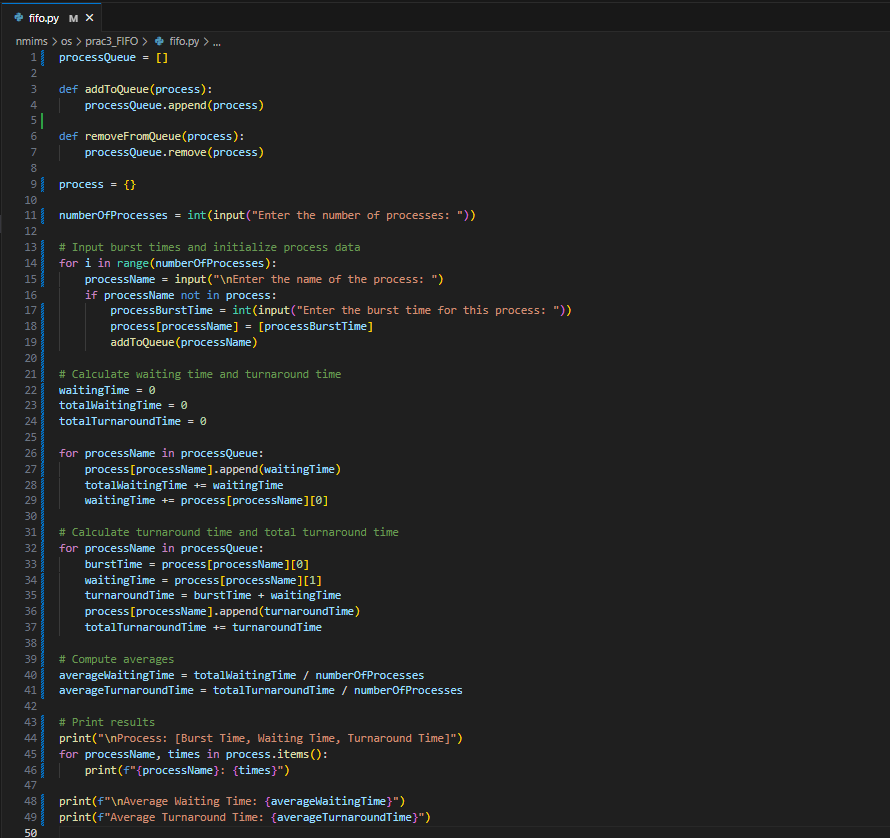
PART B

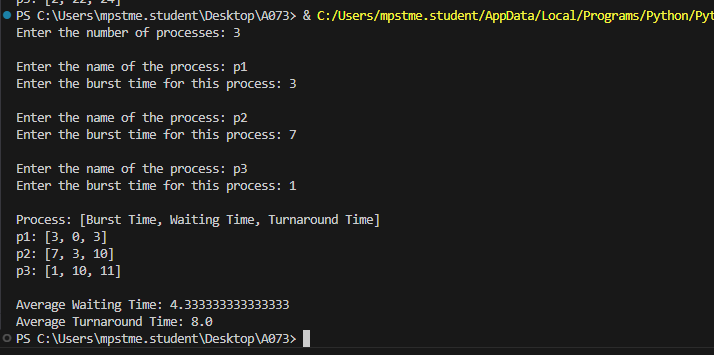
(PART B: TO BE COMPLETED BY STUDENTS)

***(Students must submit the soft copy as per following segments as per the submission instructions.)***

| Roll No.: A073 | Name: Aryan Srivastava |
| --- | --- |
| Class: MCA | Batch: B3 |
| Experiment Number- 3 | |
| Date of Experiment: | Date of Submission: 07-08-2024 |
| Grade: |  |

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

****

****

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

1. What is Process Scheduling?

Process Scheduling is a fundamental operating system function that manages the execution of processes on a computer system. It involves determining the order and allocation of CPU time to various processes in a way that optimizes performance and ensures fairness. The main goals of process scheduling are to maximize CPU utilization, ensure a good response time, and achieve high throughput.

In a multi-tasking environment, the CPU cannot process all the tasks simultaneously, so the operating system uses scheduling algorithms to decide which process runs at any given time. The process scheduler is responsible for:

1. Selecting processes from the queue of ready processes.
2. Allocating CPU time to selected processes.
3. Managing transitions between different process states

2. Why is process scheduling important in an operating system?

**Efficient CPU Utilization**: Scheduling ensures that the CPU is kept busy with productive tasks rather than being idle. By managing the order in which processes are executed, the operating system can keep the CPU as busy as possible.

**Fairness**: Scheduling algorithms aim to provide fair access to CPU time among processes. This helps prevent situations where some processes may be starved of CPU time while others monopolize it.

**Responsiveness**: For interactive systems, such as user interfaces, process scheduling is essential to ensure that user commands are processed promptly, enhancing the overall responsiveness of the system.

**Throughput**: By optimizing the process execution order, scheduling can maximize the number of processes completed in a given period, thereby increasing the system's throughput.

**Deadlines and Priorities**: In real-time systems, scheduling helps in meeting deadlines and prioritizing critical processes to ensure that time-sensitive tasks are completed within their required time frames.

**Resource Utilization**: Effective scheduling ensures that other resources, such as memory and I/O devices, are used efficiently alongside the CPU, contributing to overall system performance.

3. What are the key criteria for evaluating scheduling algorithms?

**CPU Utilization**: Measures how well the CPU is used during the execution of processes. High CPU utilization is desirable as it means the CPU is kept busy.

**Throughput**: The number of processes completed per unit of time. Higher throughput indicates that the system can handle more processes in a given time frame.

**Turnaround Time**: The total time taken for a process to complete, from submission to completion. Shorter turnaround times are generally preferred.

**Waiting Time**: The total time a process spends waiting in the ready queue before it gets CPU time. Lower waiting times contribute to better performance.

**Response Time**: For interactive systems, response time is the time from when a process starts until the first response is produced. Lower response times enhance user satisfaction.