Contents

[1. CPU-OS simulator questions. 1](#_Toc25683237)

[2. Problems that needs to be solved 2](#_Toc25683238)

# **CPU-OS simulator questions.**

Please download the CPU-OS simulator tool from the following link:

<https://drive.google.com/drive/folders/12YUK52RQ-JhP0ddj6CD_oifW4sTMbsBl?usp=sharing>

**Enter the following program in CPU-OS simulator and observe, analyse and report metrics in the sub questions with screenshots.**

**Please use the Documentation File provided to document all the results of this Assignment.**

**Steps to perform:**

* 1. Enter the source code in the compiler using compiler’s PROGRAM SOURCE[INPUT] editor window and compile the source code to generate the executable Binary Code.
  2. Load the executable code in memory.
  3. Create processes from program in the OS simulator and note down process id of each of the process that is created.
  4. Select FCFS/Round Robin/Shortest Job First/Priority scheduling algorithm.
  5. Start executing the processes.
  6. Use visualization tools to see PCB, process statistics, resources, resource utilization, Log, process states and process statistics.

Create the below Source Code [EX1.txt]:

program My\_Pgm

i = j (Value of j provided below)

for n = 1 to j

x = i \* n

next

end

1. Set the option for Round Robin to Pre-emptive. Create 3 processes P1 (j=50), P2(j=25), and P3 (j=15) from source code [EX1.txt]. Analyse the performance of Round Robin scheduler. Fill in the following table:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Process | Arrival time | Time slice | Priority | Waiting time | Turn Around Time |
| P1 | 0 | 5 | 2 | 6.26 sec | 355.1 sec |
| P2 | 1 | 2 | 3 | 353.96 sec | 473.3 sec |
| P3 | 2 | 3 | 1 | 0.14 sec | 270.1 sec |
| Average Times | | | | 73.98 sec | 366 sec |

1. Set the option for FCFS. Create 3 processes P1 (j=50), P2(j=25), and P3 (j=15) from source code [EX1.txt]. Analyse the performance of the FCFS. Fill in the following table:

|  |  |  |  |
| --- | --- | --- | --- |
| Process | Arrival time | Waiting time | Turn Around Time |
| P1 | 0 | 0.2 sec | 234.9 sec |
| P2 | 1 | 234.28 sec | 354.1 sec |
| P3 | 2 | 354.41 sec | 429.2 sec |
| Average Times | | 196.3 sec | 339.4 sec |

Draw a Gantt chart manually to illustrate the execution of these processes for the respective scheduling algorithm. Arrive at an inference on the best strategy based on the data of WAIT and Turn Around Times.

Best Algorithm is Round Robin Scheduler because it significantly reduces the wait time, and hence makes the best use of the available compute resources. FCFS does a good job reducing the average turn around time but it we would prefer lower wait time over lower turn around time.

# **Problems that needs to be solved**

Consider 5 processes whose Arrival time (AT), CPU Burst time(BT) and PRIORITY is as follows. Solve the given problem and fill up the following Table. Note: FT – Finish Time, TAT – Turn Around Time, WT – Waiting Time, RT – Response Time

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| FCFS SCHEDULING ALGORITHM | | | | | | | |
| Process | AT | BT | PRIORITY | FT | TAT | WT | RT |
| P1 | 0 | 6 | 1 | 6 | 6 | 0 | 0 |
| P2 | 4 | 2 | 1 | 8 | 4 | 2 | 2 |
| P3 | 5 | 2 | 1 | 10 | 5 | 3 | 3 |
| P4 | 6 | 5 | 1 | 15 | 9 | 4 | 4 |
| P5 | 7 | 9 | 1 | 24 | 17 | 8 | 8 |
| Average | | | |  |  |  |  |
| SJF (NON- PREEMPTIVE) SCHEDULING ALGORITHM | | | | | | | |
| Process | AT | BT | PRIORITY | FT | TAT | WT | RT |
| P1 | 0 | 6 | 1 | 6 | 6 | 0 | 0 |
| P2 | 1 | 2 | 1 | 8 | 7 | 5 | 5 |
| P3 | 5 | 2 | 1 | 10 | 5 | 3 | 3 |
| P4 | 7 | 5 | 1 | 15 | 8 | 3 | 3 |
| P5 | 7 | 9 | 1 | 24 | 17 | 8 | 8 |
| Average | | | |  |  |  |  |
| SJF (PREEMTIVE ) SCHEDULING ALGORITHM | | | | | | | |
| Process | AT | BT | PRIORITY | FT | TAT | WT | RT |
| P1 | 0 | 6 | 1 | 6 | 6 | 0 | 0 |
| P2 | 4 | 2 | 1 | 8 | 4 | 2 | 2 |
| P3 | 4 | 2 | 1 | 10 | 6 | 4 | 4 |
| P4 | 6 | 5 | 1 | 15 | 9 | 4 | 4 |
| P5 | 9 | 9 | 1 | 24 | 15 | 6 | 6 |
| Average | | | |  |  |  |  |
| ROUND ROBIN SCHEDULING ALGORITHM (TIME QUANTUM = 3) | | | | | | | |
| Process | AT | BT | PRIORITY | FT | TAT | WT | RT |
| P1 | 0 | 6 | 1 | 6 | 6 | 0 | 0 |
| P2 | 4 | 2 | 1 | 8 | 4 | 2 | 2 |
| P3 | 5 | 2 | 1 | 10 | 5 | 3 | 3 |
| P4 | 6 | 5 | 1 | 18 | 12 | 7 | 4 |
| P5 | 7 | 9 | 1 | 24 | 17 | 8 | 6 |
| Average | | | |  |  |  |  |
| PRIORITY SCHEDULING (Preemptive )ALGORITHM (least number – higher priority) | | | | | | | |
| Process | AT | BT | PRIORITY | FT | TAT | WT | RT |
| P1 | 0 | 6 | 1 | 6 | 6 | 0 | 0 |
| P2 | 4 | 2 | 1 | 8 | 4 | 2 | 2 |
| P3 | 5 | 2 | 4 | 24 | 19 | 17 | 17 |
| P4 | 6 | 5 | 3 | 22 | 16 | 11 | 11 |
| P5 | 7 | 9 | 2 | 17 | 10 | 1 | 1 |
| Average | | | |  |  |  |  |