**Submission:**

* **Submission:** You will have to submit this documentation file along-with the source code file (.sas) and zip these files with the name of the zipped file as your group number eg, **COSS\_GROUP1**.**zip**. Submit the assignment by the due date through canvas only. File submitted by any means outside CANVAS will not be accepted and marked.

In case of any issues, please drop an email to the course TAs, Ms. Michelle Gonsalves ([michelle.gonsalves@wilp.bits-pilani.ac.in](mailto:michelle.gonsalves@wilp.bits-pilani.ac.in))Please **refrain from sending emails**to center coordinators or to your faculty.

**Caution!!!**

We expect every group to submit their original work. You are not allowed to share document / other details about your assignment with other batches implementing the same problem. Any evidence of such practice will attract severe penalty.

**Evaluation:**

* The assignment carries 13 Marks
* Grading will depend on
  + Contribution of each student in the implementation of the assignment
  + **Plagiarism or copying will result in -13 marks**

## Assignment Set Number: 08 Computer Organization and Software Systems (S1-19\_DSECLZG516)

**Group Name:** [COSS\_GROUP168](https://bits-pilani.instructure.com/groups/3060)

**Contribution Table:**

1. **Contribution** (This table should contain the list of all the students in the group. Clearly mention each student’s contribution towards the assignment. Mention “No Contribution” in cases applicable.)

|  |  |  |  |
| --- | --- | --- | --- |
| **Sl. No.** | **Name(as appears in Canvas)** | **ID NO** | **Contribution** |
| **1.** | **Edward** | 2019ad04105 | **33.33%** |
| **2.** | **Sandeep** | 2019ad04106 | **33.33%** |
| **3.** | **Saurabh** | 2019ad04114 | **33.33%** |

**Solution:**

**Part 1**

\*\*Waiting Time: As per OS simulator logs

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 | BT  (minutes) | CT  (minutes) |
| P1 | 0 | 5 | 2 | 6.26sec | 355.1 sec | 5.81 | 5.81 |
| P2 | 1 | 2 | 3 | 353.96 sec | 473.3 sec | 1.98 | 7.80 |
| P3 | 2 | 3 | 1 | 0.14 sec | 270.1 sec | 4.50 | 12.29 |
| Average Times | | | | 73.98 sec | 366 sec |  |  |

1. **Gantt Chart**

|  |  |  |
| --- | --- | --- |
| P1 | P2 | P3 |
| 0 5.81 | 7.80 | 12.29 |

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 | BT  (minutes) | CT  (minutes) |
| P1 | 0 | 0.2 sec | 234.9 sec | 3.9 | 3.9 |
| P2 | 1 | 234.28 sec | 354.1 sec | 2.0 | 5.9 |
| P3 | 2 | 354.41 sec | 429.2 sec | 2.3 | 8.2 |
| Average Times | | 196.3 sec | * 1. sec |  |  |

**B. Gantt chart**

|  |  |  |
| --- | --- | --- |
| P1 | P2 | P3 |
| 0 3.9 | 5.9 | 8.20 |

**Conclusion**: Based on Waiting Time and Turn-Around-Time, the process gets completed faster in FCFS algorithm. Hence, FCFS is the best strategy.

**Part 2: Solved**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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 | | | | | 12.6 | 8.2 | 3.4 | 3.4 |
|  | |  |  |  |  |  |  |  |
| 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 | | | | | 12.6 | 8.6 | 3.8 | 3.8 |
|  | |  |  |  |  |  |  |  |
| 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 | | | | | 12.6 | 8 | 3.2 | 3.2 |
|  | |  |  |  |  |  |  |  |
| 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 | | | | | 13.2 | 8.8 | 4 | 3 |
|  | |  |  |  |  |  |  |  |
| PRIORITY SCHEDULING (Pre-emptive ) 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 | | | | | 15.4 | 11 | 6.2 | 6.2 |

---End of assignment---