

CSN6214 Operating Systems

Trimester 2430

Group Assignment

Instructions:

- a) Form a group of **FOUR (4)** members.
 - Each group must have four members from the same tutorial group
- b) You are allowed to use any programming language to simulate these concepts, not limited to C++, Java, Python, etc.
- c) The group leader should submit the working program or prototype on eBwise by **31 January 2025, 11:59 pm**. **Any late submission will be deducted 10% from the awarded mark for each day late.**
- d) All the team members are required to present their work to the lecturer during the given slots in Week 14, start from **03 February 2024 to 09 February 2024**. The presentation will be conducted physically.
- e) Any form of plagiarism will not be tolerated. No marks will be given for this assignment if plagiarism is detected.

Topic 1:

- Simulation of CPU scheduling algorithms
- Process Scheduling algorithms: Part A is compulsory to execute, and choose any 3 scheduling algorithms from Part B.

Part A

- a) Round Robin with Quantum 3

Part B

- b) SRT
 - c) SJN
 - d) Preemptive Priority
 - e) Non-Preemptive Priority
-
- f) Users should be able to enter details about the processes such as Arrival Time, Burst Time, Priority, Time Quantum for Round Robin assigned at the beginning of simulation and the number of processes can range from 3 to 10.
 - g) Executing the program should show the Gantt chart (visual form) of each algorithm.

h) Calculation of

- i. Turnaround time for each process
- ii. Total and Average Turnaround time for the entire processes
- iii. Waiting time for each process
- iv. Total and Average Waiting time for the entire processes

Pre-assigned case

A sample of case study listed below for students to test the program, but lecturers will choose other case study during the presentation of prototype.

Process	Burst time	Arrival time	Priority
P0	6	0	3
P1	4	1	3
P2	6	5	1
P3	6	6	1
P4	6	7	5
P5	6	8	6

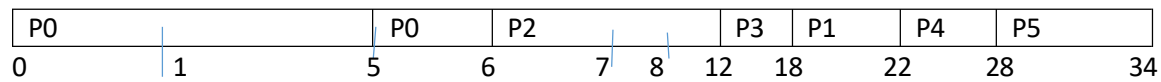
Assumption

- I. For Preemptive Priority, if the current process (in CPU) has SAME PRIORITY with new arrival process, NO PREEMPTION will be executed, TO REDUCE CONTEXT SWITCH
- II. For Preemptive Priority, if the current process (in CPU) has SAME Burst time with new arrival process, NO PREEMPTION will be executed, TO REDUCE CONTEXT SWITCH
- III. For Round Robin, if there are two or more processes with SAME PRIORITY and same arrival time in the ready poll, privilege go to the process that not being executed before (to give chance to "hungry" process to be fed into CPU).
- IV. For all algorithms, if SAME PRIORITY for those processes in the ready poll, CHOOSE FCFS
- V. For all algorithms, if BURST TIME SAME for those processes in the ready poll, CHOOSE FCFS
- VI. For all algorithms, the lower the priority counts, the higher the priority for a process

Expected output:

Sample inputs and outputs:

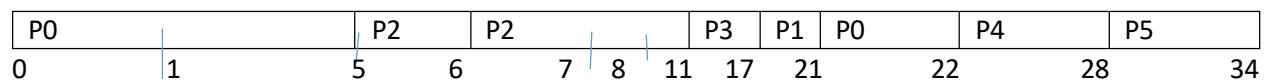
i. NON-Preemptive Priority



SAME PRIORITY, NO PREEMPTION, TO REDUCE CONTEXT SWITCH

SAME PRIORITY, CHOOSE FCFS

ii. Preemptive Priority

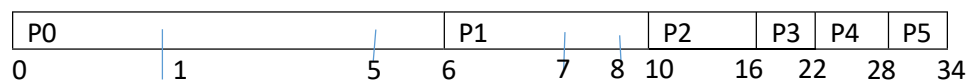


SAME PRIORITY, NO PREEMPTION, TO REDUCE CONTEXT SWITCH

SAME PRIORITY, CHOOSE FCFS

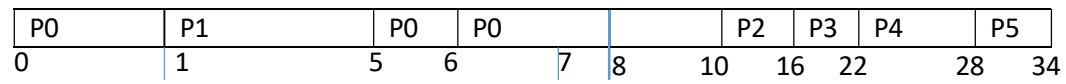
	ARRIVAL TIME	BURST TIME	FINISHING TIME	TURNAROUND TIME	WAITING TIME
P0	0	6	22	22	16
P1	1	4	21	20	16
P2	5	6	11	6	0
P3	6	6	17	11	5
P4	7	6	28	21	15
P5	8	6	34	26	20

iii. SJN



- IF BURST TIME SAME, CHOOSE FCFS

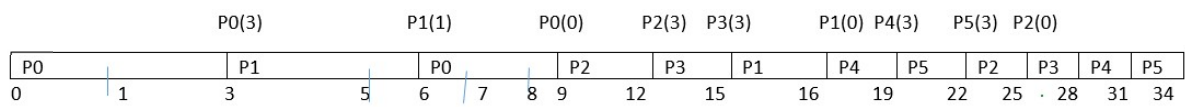
iv. SRT



- If same burst time, look for FCFS

v. Round Robin with Quantum 3

Round Robin with Quantum 3



P0(6) P1(4) P2(6) P3(6) P4(6) P5(6)

	ARRIVAL TIME	burst TIME	FINISHING TIME	TURNAROUND TIME	WAITING TIME
P0	0	6	9	9	3
P1	1	4	16	15	11
P2	5	6	25	20	14
P3	6	6	28	22	16
P4	7	6	31	24	18
P5	8	6	34	26	20