

MA 2233 Data Structures and Applications Lab

Quiz 3 - Priority Queue

02.12.2021

We know that a priority queue is a data structure that is typically used by a scheduler when it needs to determine the next job/task to be performed based on priority. In Quiz I on 04.10.2021, you were expected to implement a simple priority queue for the following list of jobs submitted at the time $t=0$. The jobs can be at 3 different levels of priority,

Time Submitted	$t = 0$	0	0	0	0	0
Job No	1	2	3	4	5	6
Priority	II	III	I	II	I	II
Time Required for Completion (TRC)	30	20	40	10	25	20

with a scheduler implementing the following policy:

[P1] Jobs of higher priority in the queue are always executed first.

[P2] When two jobs of the same priority are submitted the one needing more time to complete is given more preference.

[P3] When two jobs of the same priority and identical time-for-completion are submitted the one that has been in the queue for a longer time is given more preference. Ties are broken arbitrarily.

[P4] Once a job is committed to the processor, till it is completed the scheduler does not load the next job.

Accordingly, we obtained the following completion time-lines for the above jobs:

Job No	3	5	1	6	4	2
Timeline	1-40	41-65	66-95	96-115	116-125	126-145

Now, consider that new jobs are added at different time instants ' t ' to the queue with the following priorities and TRC:

Time Submitted	$t = 25$	60	70	150	190	210
Job No	7	8	9	10	11	12
Priority	II	I	III	II	I	II
TRC	35	30	30	20	20	20

Assuming that jobs are added to the queue parallelly and the time to load a job onto the processor is negligible, we obtain the following completion time-lines for the above jobs:

Job No	3	5	8	7	1	6	10	11	12	4	9	2
Timeline	1-40	41-65	66-95	96-130	131-160	161-180	181-200	201-220	221-240	241-250	251-280	281-300

What needs to be done?

Q1: Needs coding and implementation:

- Implement a priority queue for the above modified scheduling policy and validate your results with the given data.
- You could use your earlier implementation of PQ and modify the scheduling policy accordingly.

Q2: No need to code. Answer in the body of your response e-mail.

- What changes are required to your code from your earlier implementation, assuming your original implementation was working correctly.
- What changes are required to your code if the scheduling policy **[P4]** is changed to **[P4']** as follows:

[P4'] = Modified [P4]:

- Even after a job is committed to the processor, at every time instant 't', the scheduler looks for jobs that might have been added to the queue with a higher priority.
- It swaps out the current job that is being executed with the new job of higher priority.
- Sends the incomplete task to be added back to the priority queue.

- Does it also lead to any other modifications / deletions to the overall scheduling policy?