Great Place To Work®
Certified
FEB 2023 – FEB 2024
INDIA

UPES

UNIVERSITY OF TOMORROW

nirf
RANKING 2023

#RANKED 52
IN INDIA

#University Category

QS WORLD UNIVERSITY RANKINGS

NO.1 PVT. UNIVERSITY IN ACADEMIC REPUTATION IN INDIA

ACCREDITED WITH GRADE A NAAC

ACCREDITED **GRADE 'A'** BY NAAC

E-LEAD

PERFECT SCORE OF **150/150** AS A TESTAMENT TO EXECEPTIONAL E-LEARNING METHODS

Unit 2 : Process and Thread Management

Lecture 4

Submitted by:

Dr Khushboo Jain

School of Computer Science
UPES, Dehradun
India

Table of Contents

1. Non Preemptive Priority scheduling and Preemptive Scheduling,
2. Priority Round Robin
3. Multilevel Queue
4. Multiple Feedback Queue

Learning & Course Outcomes

***LO1:** Define Non-Preemptive Priority Scheduling and Preemptive Scheduling.*
***LO2:** Describe how Priority Round Robin, Multilevel Queue, and Multiple Feedback Queue scheduling algorithms work.*
***LO3:**Apply Non-Preemptive Priority Scheduling and Preemptive Scheduling algorithms to solve scheduling problems.*

***CO2:** Evaluate and analyze process and thread scheduling techniques, discerning their benefits and challenges.*

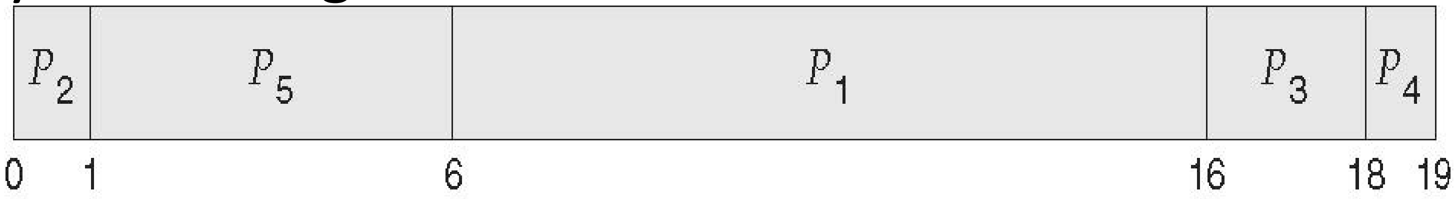
Priority Scheduling

- A priority number (integer) is associated with each process
- The CPU is allocated to the process with the highest priority (smallest integer ≡ highest priority)
    - Preemptive
    - Nonpreemptive
- SJF is priority scheduling where priority is the inverse of predicted next CPU burst time
- Problem ≡ **Starvation** – low priority processes may never execute
- Solution ≡ **Aging** – as time progresses increase the priority of the process

Example of Priority Scheduling

Process	Burst Time	Priority
P_1	10	3
P_2	1	1
P_3	2	4
P_4	1	5
P_5	5	2

- Priority scheduling Gantt Chart



- Average waiting time = 8.2 msec

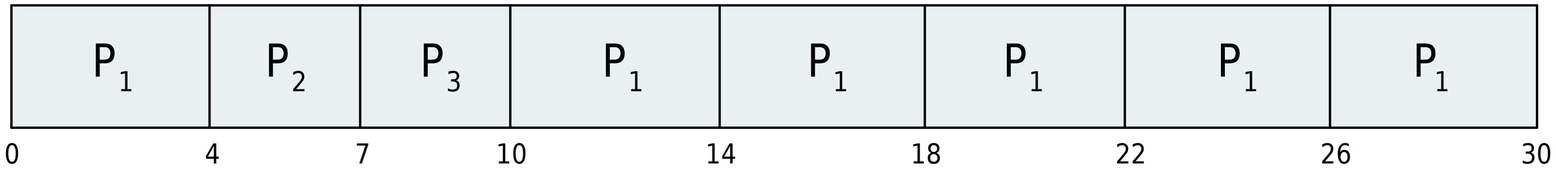
Round Robin (RR)

- Each process gets a small unit of CPU time (**time quantum** *q*), usually 10-100 milliseconds. After this time has elapsed, the process is preempted and added to the end of the ready queue.
- If there are *n* processes in the ready queue and the time quantum is *q*, then each process gets 1/*n* of the CPU time in chunks of at most *q* time units at once. No process waits more than $(n-1)q$ time units.
- Timer interrupts every quantum to schedule next process
- Performance
  - *q* large $\Rightarrow$ FIFO
  - *q* small $\Rightarrow$ *q* must be large with respect to context switch, otherwise overhead is too high

Example of RR with Time Quantum = 4

Process	Burst Time
P_1	24
P_2	3
P_3	3

- The Gantt chart is:



- Typically, higher average turnaround than SJF, but better ***response***
- q should be large compared to context switch time
- q usually 10ms to 100ms, context switch < 10 usec

Time Quantum and Context Switch Time

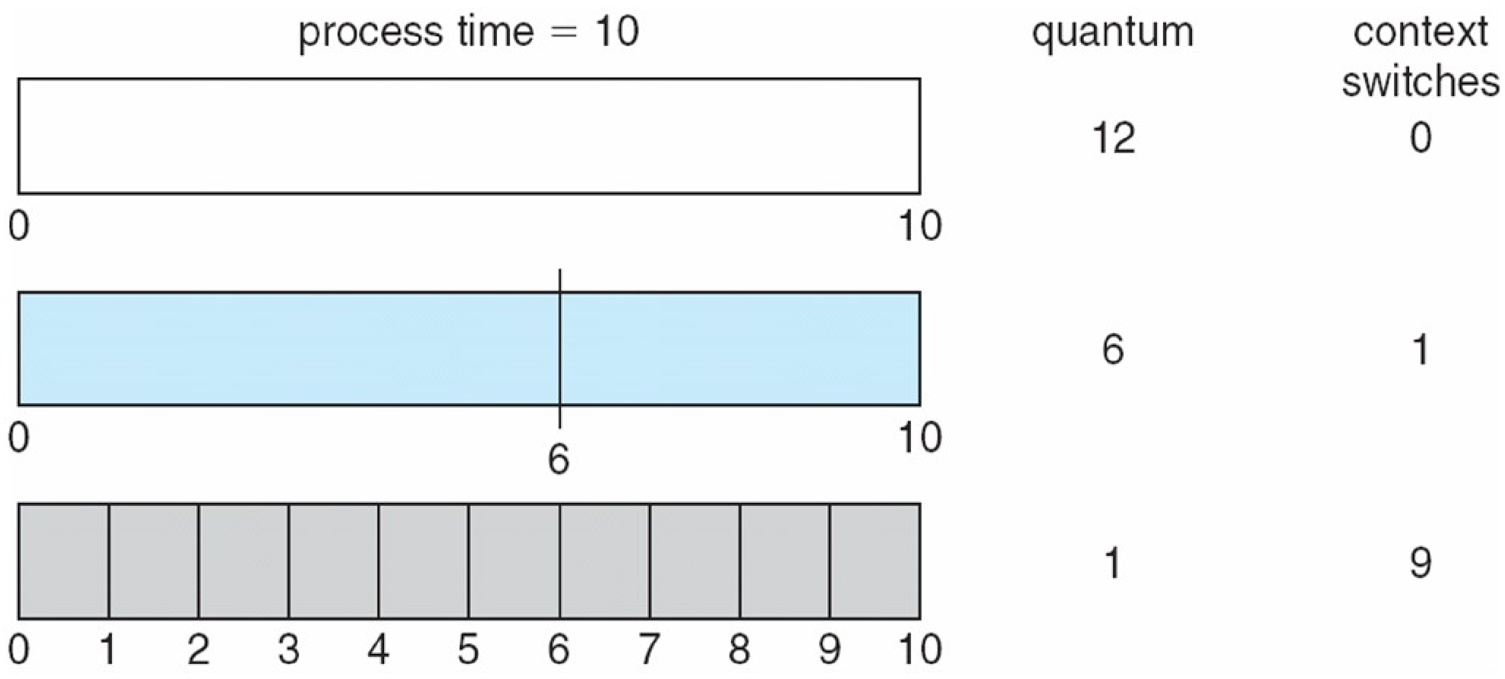
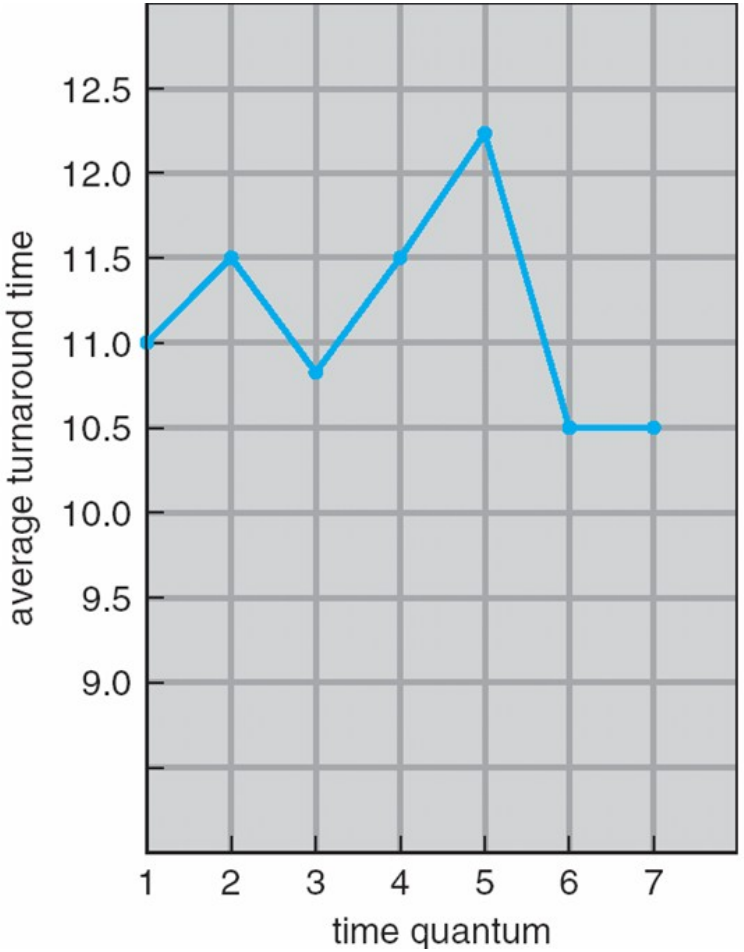


Fig 1. Time Quantum and Context Switch Time [1]

Turnaround Time Varies With The Time Quantum



process	time
P_1	6
P_2	3
P_3	1
P_4	7

80% of CPU bursts
should be shorter than q

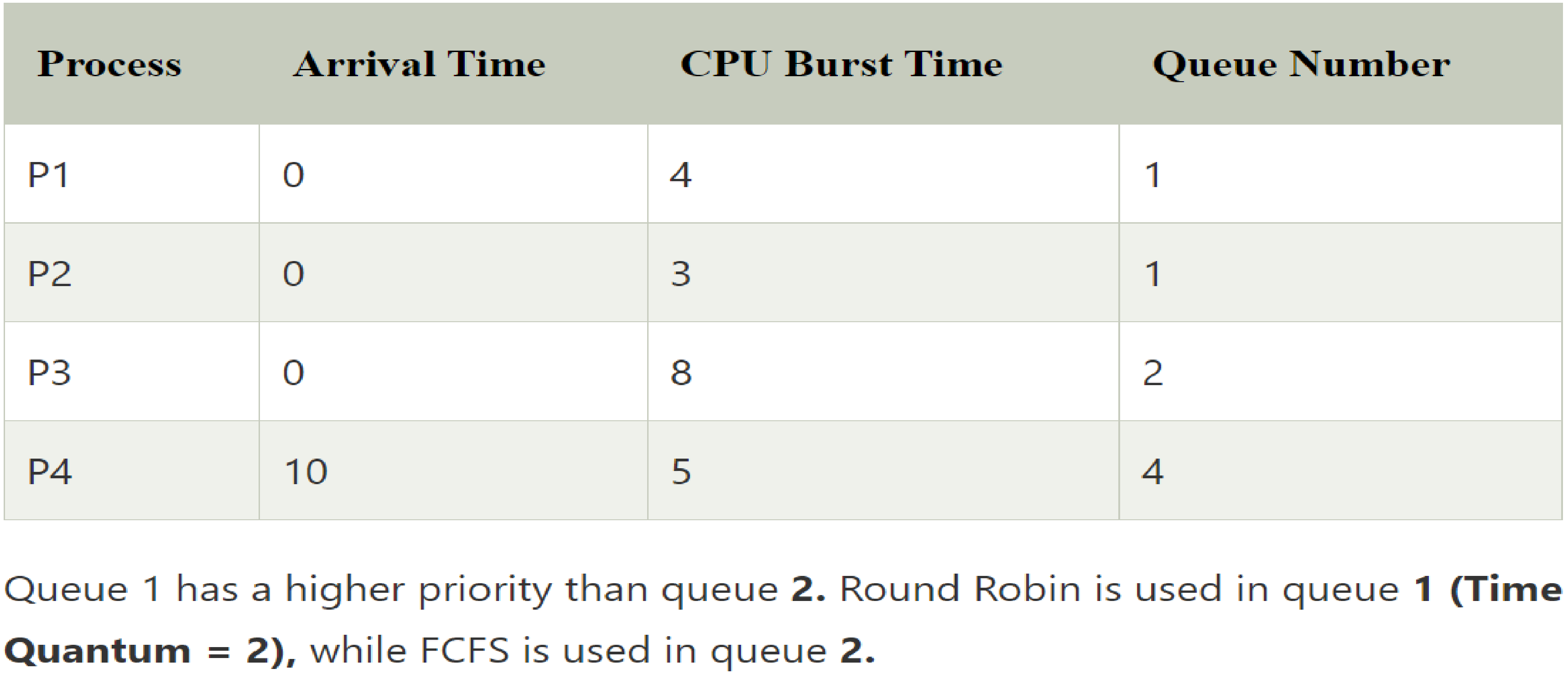
Fig 2. Turnaround Time Varies With The Time Quantum [1]

Multilevel Queue

- Ready queue is partitioned into separate queues, eg:
  - **foreground** (interactive)
  - **background** (batch)
- Process permanently in a given queue
- Each queue has its own scheduling algorithm:
  - foreground – RR
  - background – FCFS
- Scheduling must be done between the queues:
  - Fixed priority scheduling; (i.e., serve all from foreground then from background). Possibility of starvation.
  - Time slice – each queue gets a certain amount of CPU time which it can schedule amongst its processes; i.e., 80% to foreground in RR
  - 20% to background in FCFS

Numerical on Multilevel Queue

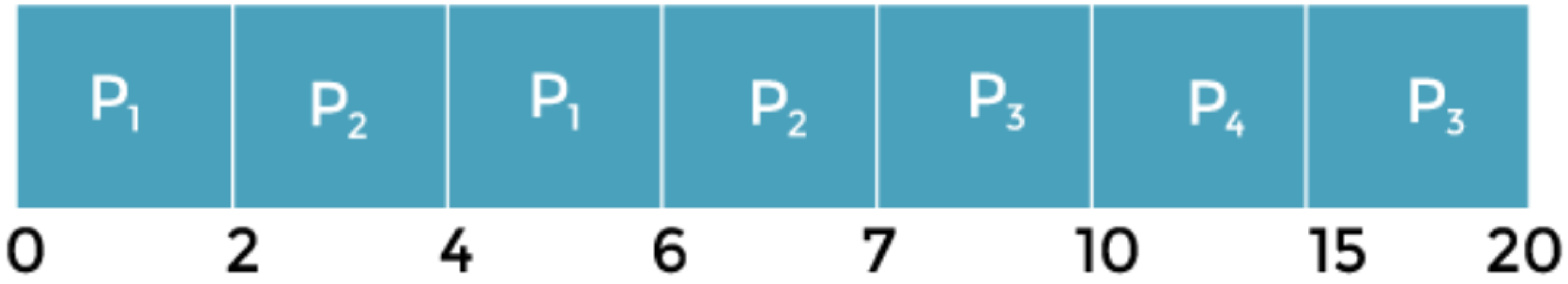
- Let’s take an example of a multiple queue-scheduling (MQS) algorithm that shows how the multilevel queue scheduling work. Consider the four processes listed in the table below under multilevel queue scheduling. The queue number denotes the process’s queue.



Process	Arrival Time	CPU Burst Time	Queue Number
P1	0	4	1
P2	0	3	1
P3	0	8	2
P4	10	5	4

Queue 1 has a higher priority than queue **2.** Round Robin is used in queue **1 (Time Quantum = 2),** while FCFS is used in queue **2.**

Numerical on Multilevel Queue



Working

1. Both queues have been processed at the start. Therefore, **queue 1(P1, P2)** runs first (due to greater priority) in a round robin way and finished after 7 units.
2. The process in queue 2 (P3) starts running (since there is no process in queue 1), but while it is executing, P4 enters queue 1 and interrupts P3 and then P3 takes the CPU and finishes its execution.

Multilevel Queue Scheduling

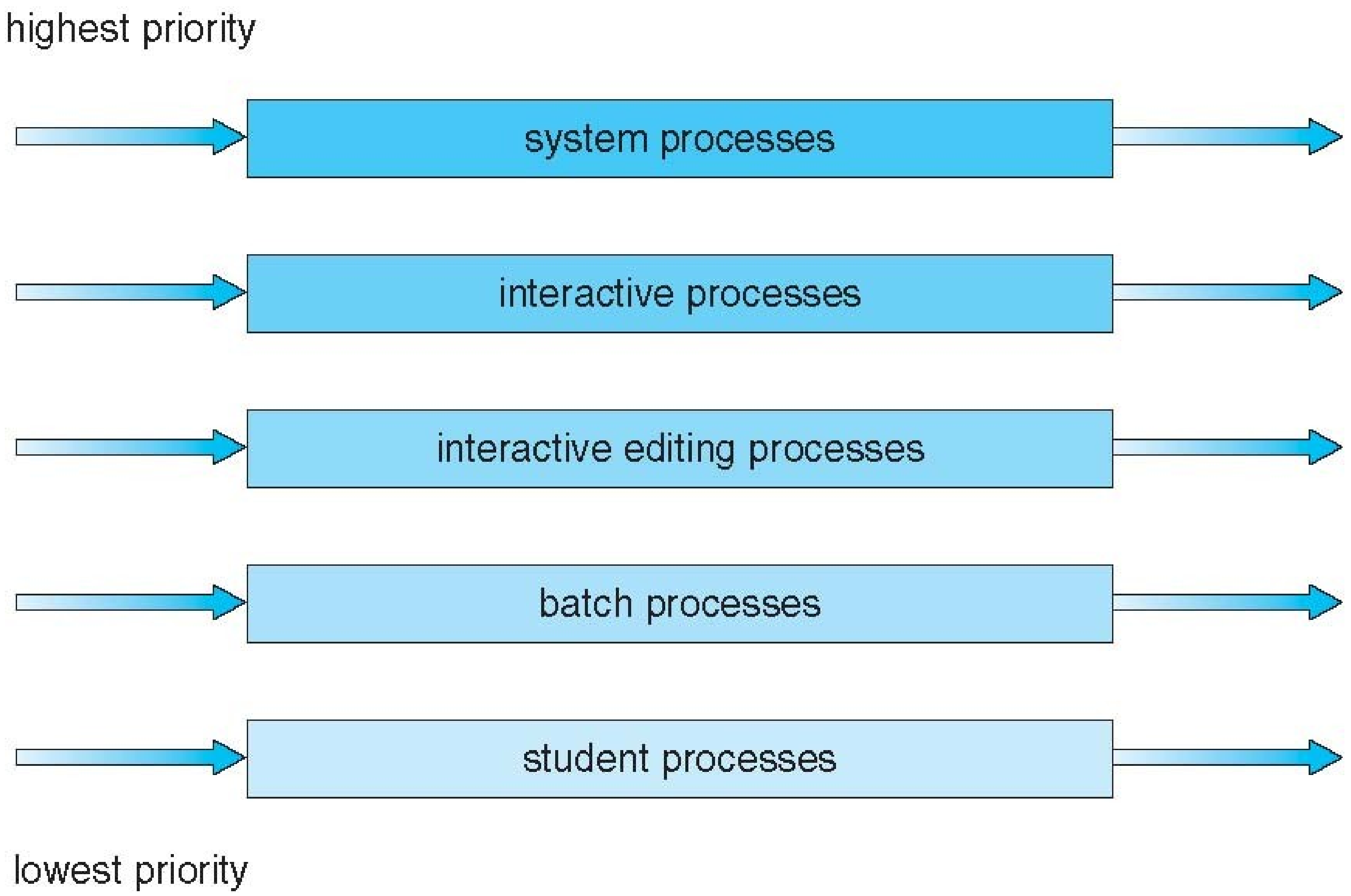


Fig 3. Multilevel Queue Scheduling [1]

Multilevel Feedback Queue

- A process can move between the various queues; aging can be implemented this way
- Multilevel-feedback-queue scheduler defined by the following parameters:
  - number of queues
  - scheduling algorithms for each queue
  - method used to determine when to upgrade a process
  - method used to determine when to demote a process
  - method used to determine which queue a process will enter when that process needs service

Example of Multilevel Feedback Queue

Three queues:
Q0 – RR with time quantum 8 milliseconds
Q1 – RR time quantum 16 milliseconds
Q2 – FCFS

Scheduling
A new job enters queue Q0 which is served FCFS
When it gains CPU, job receives 8 milliseconds
If it does not finish in 8 milliseconds, job is moved to queue Q1
At Q1 job is again served FCFS and receives 16 additional milliseconds
If it still does not complete, it is preempted and moved to queue Q2

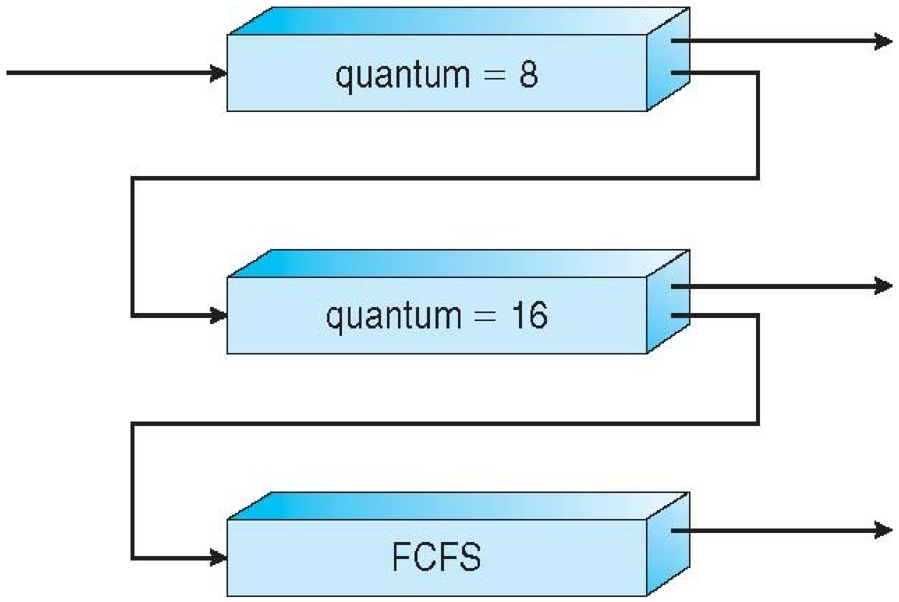


Fig 4. Example of multilevel feedback queue [1]

Numerical of Multilevel Feedback Queue

Example: Consider a system that has a CPU-bound process, which requires a burst time of 40 seconds. The multilevel Feed Back Queue scheduling algorithm is used and the queue time quantum ‘2’ seconds and in each level it is incremented by ‘5’ seconds. Then how many times the process will be interrupted and in which queue the process will terminate the execution?

Solution:

Process P needs 40 Seconds for total execution.
At Queue 1 it is executed for 2 seconds and then interrupted and shifted to queue 2.
At Queue 2 it is executed for 7 seconds and then interrupted and shifted to queue 3.
At Queue 3 it is executed for 12 seconds and then interrupted and shifted to queue 4.
At Queue 4 it is executed for 17 seconds and then interrupted and shifted to queue 5.
At Queue 5 it executes for 2 seconds and then it completes.
Hence the process is interrupted 4 times and completed on queue 5.

Summary

- **Non-Preemptive Priority Scheduling:** Processes are executed based on priority without interruption. Higher priority processes are selected first, and once a process starts, it runs to completion.
- **Preemptive Priority Scheduling:** Processes are executed based on priority, but can be interrupted by higher priority processes. Ensures that the highest priority process always runs, improving responsiveness.
- **Priority Round Robin:** Combines priority scheduling with round-robin scheduling. Processes are grouped by priority levels and within each priority, round-robin scheduling is applied.
- **Multilevel Queue Scheduling:** Multiple queues for different types of processes with distinct scheduling algorithms for each queue. Processes are permanently assigned to a queue based on specific criteria such as priority or type.
- **Multiple Feedback Queue Scheduling:** Processes can move between queues based on their behavior and execution history. Aims to dynamically adjust process priorities to optimize performance and responsiveness.

Reference Material

[1]. Silberschatz, A. & Galvin, P. (2009) Operating System Concepts. 8th ed. NJ: John Wiley & Sons, Inc.

- Download Link- https://www.mbit.edu.in/wp-content/uploads/2020/05/Operating_System_Concepts_8th_EditionA4.pdf

[2]. NPTEL Video Lecture: https://onlinecourses.nptel.ac.in/noc24_cs80/preview

UPES
UNIVERSITY OF TOMORROW

MCQ’s

1. Which of the following scheduling algorithms allows a process to run until it voluntarily relinquishes control of the CPU?

   a) Non-Preemptive Priority Scheduling
   b) Preemptive Scheduling
   c) Priority Round Robin
   d) Multilevel Queue

2. In preemptive scheduling, when does a context switch occur?

   a) When a process finishes execution
   b) When a process voluntarily releases the CPU
   c) When a higher-priority process becomes ready to run
   d) When a process is waiting for I/O

MCQ’s

3. Which scheduling algorithm assigns a priority to each process and runs the highest priority process first?

a) Priority Round Robin
b) Multilevel Queue
c) Multiple Feedback Queue
d) Non-Preemptive Priority Scheduling

4. In Priority Round Robin scheduling, how is the time quantum determined for each priority level?

a) It is the same for all priority levels.
b) It is dynamically adjusted based on process behavior.
c) It is based solely on the priority of the process.
d) It is set by the system administrator.

MCQ’s

5. Which scheduling algorithm uses multiple queues to separate processes based on their priority levels?

   a) Multilevel Queue
   b) Non-Preemptive Priority Scheduling
   c) Preemptive Scheduling
   d) Multiple Feedback Queue

6. In a Multilevel Queue scheduling algorithm, how are processes assigned to different queues?

   a) Based on the execution time of the process
   b) Based on the arrival time of the process
   c) Based on the process priority
   d) Based on the process ID

MCQ’s

7. Which scheduling algorithm allows a process to move between different queues based on its behavior?

a) Non-Preemptive Priority Scheduling
b) Preemptive Scheduling
c) Multiple Feedback Queue
d) Priority Round Robin

8. What is the main advantage of using a Multiple Feedback Queue scheduling algorithm?

a) It is simpler to implement compared to other algorithms.
b) It ensures fairness by giving each process an equal share of the CPU.
c) It allows processes to have varying levels of priority.
d) It adapts well to both interactive and CPU-bound processes.

MCQ’s

9. Which scheduling algorithm combines the features of both preemptive and non-preemptive scheduling?

a) Priority Round Robin
b) Multilevel Queue
c) Multiple Feedback Queue
d) Preemptive Scheduling

MCQ’s Answers

Answers:
1.a) Non-Preemptive Priority Scheduling
2.c) When a higher-priority process becomes ready to run
3.d) Non-Preemptive Priority Scheduling
4.c) It is based solely on the priority of the process.
5.a) Multilevel Queue
6.c) Based on the process priority
7.c) Multiple Feedback Queue
8.d) It adapts well to both interactive and CPU-bound processes.
9.a) Priority Round Robin

What’s Next

1. Real Time scheduling :
   1. Rate Monotonic
   2. Earliest Deadline First
2. Operations on processes:
   1. Creation and Termination

UPES
UNIVERSITY OF TOMORROW

Thank You