CMPT 300
Operating System I
CPU Scheduling – Chapter 5

Dr. Hazra Imran

Summer 2022

Round-Robin Scheduling

5 ms 15 ms

- Adding time-based preemption to FCFS scheduling produces **round-robin** (RR) scheduling
  - Processes get a fixed-size **time slice** or **time quantum** on CPU
- Again, process ready-queue is a simple FIFO
  - Current process runs until it blocks, yields or terminates, or until it has used up its entire time slice.
  - When a process is moved off the CPU, it is put at end of run queue
  - Next process to receive the CPU is taken from front of the queue

System perfo — how large time slice is.

Example of RR with Time Quantum = 4

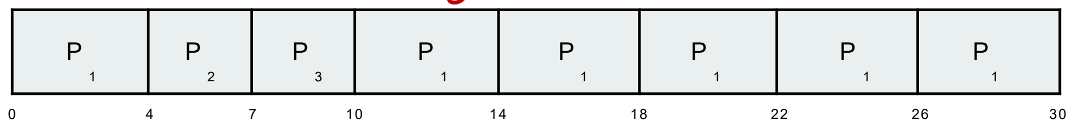
	Process	Burst Time
4	P_1	24
4	P_2	3
4	P_3	3

The Gantt chart is: ?

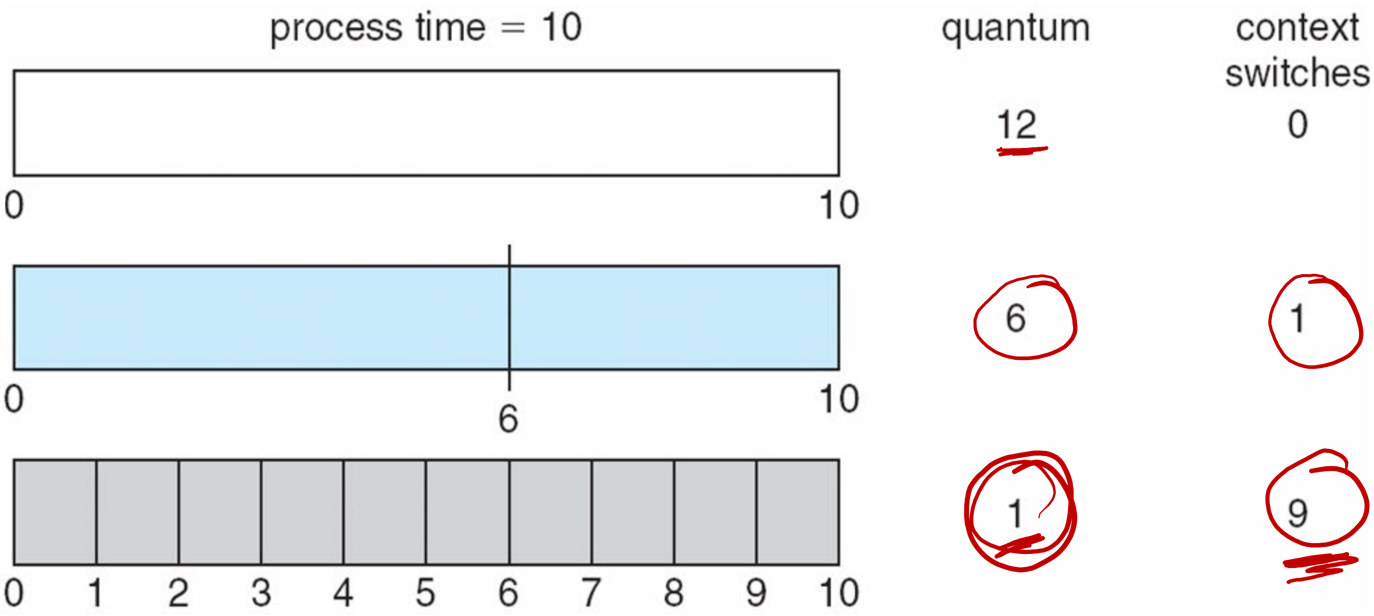
Example of RR with Time Quantum = 4

Process	Burst Time
P_1	24
P_2	3
P_3	3

The Gantt chart is: ?



Time Quantum and Context Switch Time



quantum	context switches
12	0
6	1
1	9

Multilevel Queue Scheduling

- Processes can often be categorized based on their purpose and behavior, e.g.
  - System processes
  - Interactive processes
  - Interactive editing processes
  - Batch processes

- Additionally, divide processes into two main categories: foreground processes and background processes
  - **Foreground processes** need responsiveness, have small CPU bursts
  - **Background processes** have large CPU bursts, and aren't interactive

Multilevel Queue Scheduling

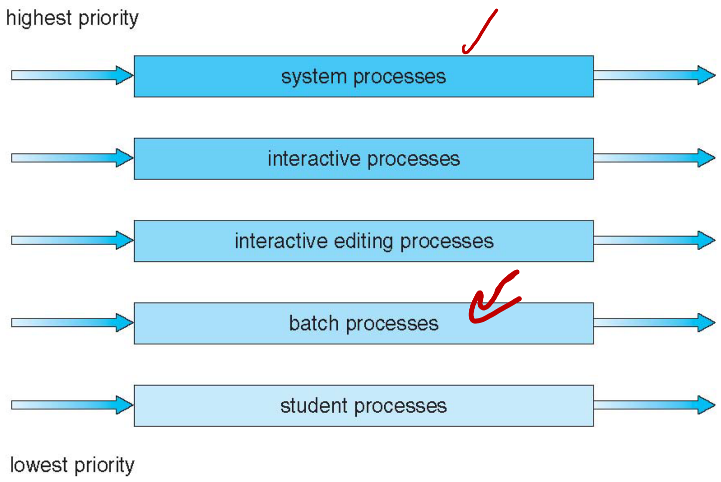
- **Multilevel queue scheduling** maintains a queue for each category of process
  - Queues have a decreasing priority – e.g. system processes are highest priority, batch processes are lowest priority
  - Processes are permanently assigned to a specific queue when they are started, and are not moved between different queues

Multilevel Queue Scheduling

- Each queue has its own fixed priority

- Usually, high-priority queues always preempt low-priority
  - As long as there are system processes ready to run, they run first!
  - Interactive processes only run when no system processes can run
  - etc.
  - Batch processes only run if no other processes are ready to run

Also possible to divide CPU time across subset of queues
e.g. spend 80% of CPU time running interactive processes, 20% running batch processes



Multilevel Queue Scheduling

- Each queue can also have its own scheduling algorithm and parameters (e.g. time-slice size)
    - Batch processes can be run with first-come first-served scheduling, or round-robin with a very large time-slice (for runaway processes)
    - Other processes typically run with round-robin scheduling

Multilevel Feedback Queue Scheduling

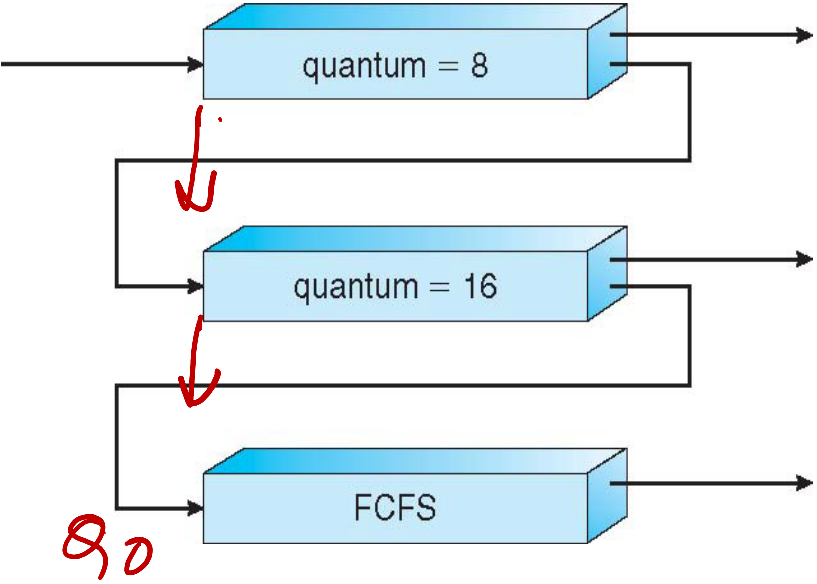
- **Multilevel feedback queue** scheduling allows processes to move between the different priority queues

- 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:
  - Q_0 – RR with time quantum 8 milliseconds
  - Q_1 – RR time quantum 16 milliseconds
  - Q_2 – FCFS

- Scheduling
  - A new job enters queue Q_0 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 Q_1
  - At Q_1 job is again served FCFS and receives 16 additional milliseconds
    - If it still does not complete, it is preempted and moved to queue Q_2



Multilevel Feedback Queues

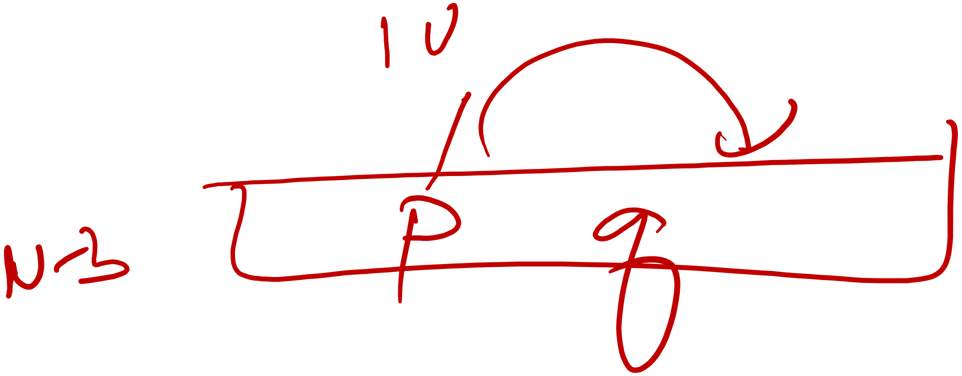
- Mac OS X has multiple queues for threads, falling into four priority bands:
  - Normal (lowest priority), system high priority, kernel mode only, real-time threads (highest priority)
- Solaris uses 170 queues, divided into various categories
- Linux used a multilevel feedback queue

Clicker

Process p is in the queue at level N-3, followed by process q at the same level. Queues at levels N through N-2 are empty. The time slice is 1 unit.

If p needs 2 units of CPU time and q needs 1 unit of CPU time, process __________ will terminate first.

(A) p [✓ (B) q]



Clicker

When p starts executing, a new process r with a CPU time requirement of 3 units arrives at level N. The 3 processes will terminate in the order __________.

(A) r, p, q

[✓ (B) r, q, p]

(C) p , q, r

p = 2

q = 1

r = 3

HP r

p, q

Next

Ch 6 - Synchronization tool