PES UNIVERSITY ONLINE

OPERATING SYSTEMS

CPU Scheduling _ 3

Nitin V Pujari
Faculty, Computer Science
Dean - IQAC, PES University

OPERATING SYSTEMS

- Preemptive Priority Scheduling (PPS)
- Round Robin Scheduling (RR)
- Multilevel Queues
- Multilevel Feedback Queue

Nitin V Pujari
Faculty, Computer Science
Dean - IQAC, PES University

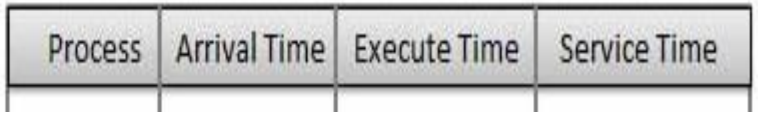
Scheduling Criteria

- **CPU utilization** – keep the CPU as busy as possible
- **Throughput** – # of processes that complete their execution per time unit
- **Turnaround time** – amount of time to execute a particular process (performance metric)
- **Waiting time** – amount of time a process has been waiting in the ready queue
- **Response time** – amount of time it takes from when a request was submitted until the first response is produced, not output (for time-sharing environment)

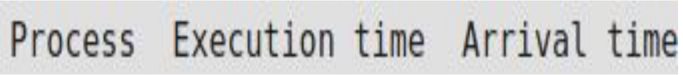
Scheduling Algorithm Optimization Criteria

- Maximize CPU utilization
- Maximize throughput
- Minimize turnaround time
- Minimize waiting time
- Minimize response time

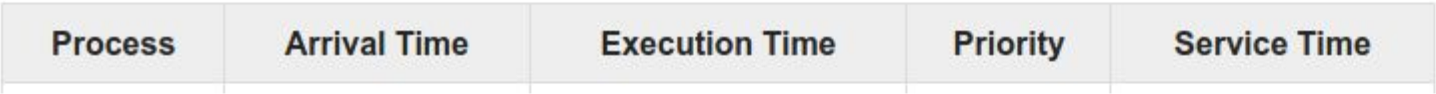
Sample Schemas and Tools for solving Scheduling Problems



Process	Arrival Time	Execute Time	Service Time

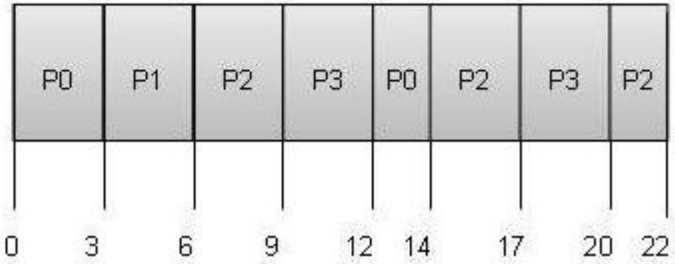


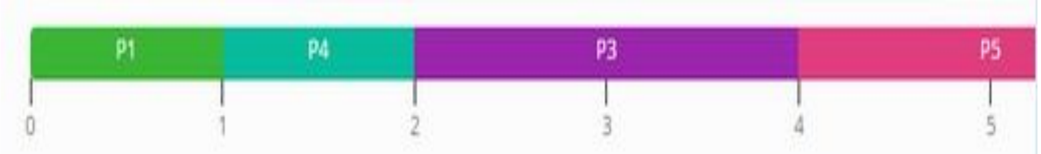
Process	Execution time	Arrival time

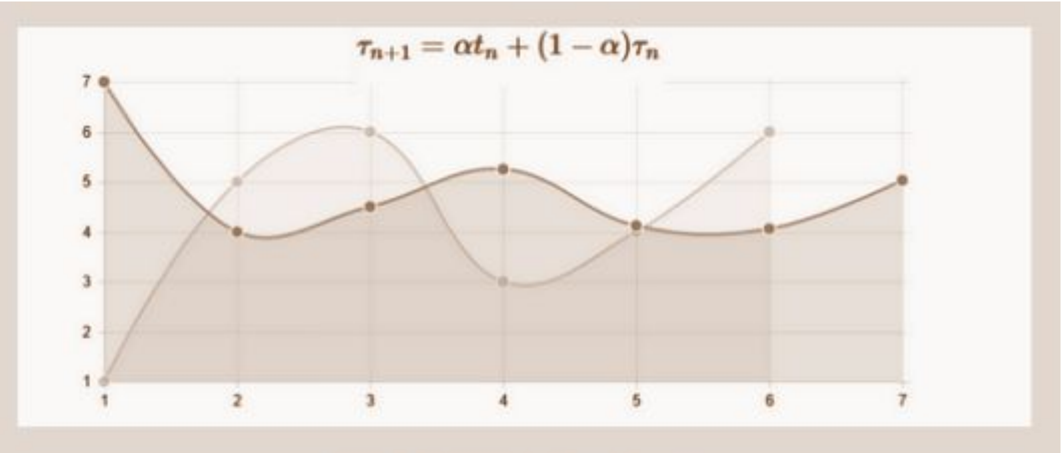


Process	Arrival Time	Execution Time	Priority	Service Time

GANTT Chart







CPU Burst Predictor

Sample Schemas and Tools for solving Scheduling Problems

Arrival Time: Time at which the process arrives in the ready queue.

Completion Time: Time at which process completes its execution.

Burst Time: Time required by a process for CPU execution.

Turn Around Time: Time Difference between completion time and arrival time.

Turn Around Time = Completion Time – Arrival Time

Waiting Time(W.T): Time Difference between turn around time and burst time.

Waiting Time = Turnaround Time – Burst Time

Preemptive Priority Scheduling (NPPS)

Characteristics of Preemptive Priority Scheduling

A CPU algorithm that schedules processes based on priority.

- In **Preemptive Priority Scheduling**, at the time of arrival of a process in the ready queue, its Priority is compared with the priority of the other processes present in the ready queue as well as with the one which is being executed by the CPU at that point of time.
- The **One with the highest priority** among all the available processes will be given the **CPU next,** whenever the decision is to be made
- The difference between **preemptive priority scheduling** and n**on preemptive priority scheduling** is that, in the preemptive priority scheduling, the job which is being executed can be **stopped** at the **arrival** of a **higher priority job**
- Once all the jobs get available in the ready queue, the algorithm will behave as non-preemptive priority scheduling, which means the job scheduled will run till the completion and no preemption will be done.

Preemptive Priority Scheduling (PPS)

Advantages of Preemptive Priority Scheduling

- Priority scheduling is simple to understand.
- Priority scheduling is a user-friendly algorithm.
- Based on priority, processes are executed. So, the process which has the highest priority does not need to wait for more time
- Priorities in the Priority scheduling are chosen on the basis of time requirements, memory requirements, and user preferences.

Disadvantages of Preemptive Priority Scheduling

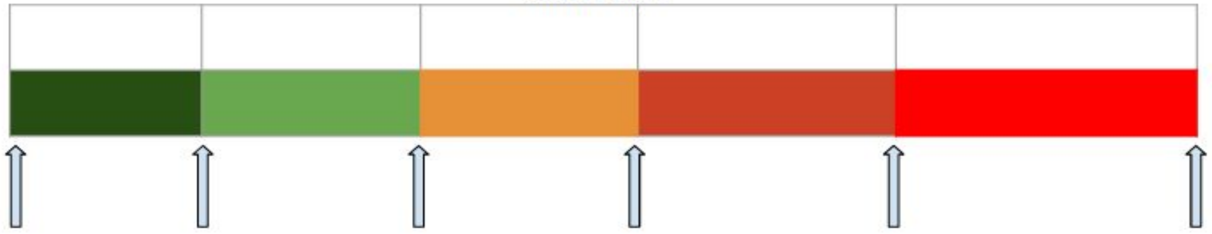
- When the multiple processes have the same priorities, then we have to apply another scheduling algorithm.
- In priority scheduling, if the system is crashed, then all low-priority processes that are not yet completed will also get lost.
- Another disadvantage of Priority Scheduling is starvation. The problem of starvation is a situation that arises when a process does not get the required resources like CPU because another process is holding the resources, and it waits for a long time for the CPU.
- ‘Aging’ is a technique that is used to remove the problem of starvation.
- In aging, the system automatically increases the priorities of the processes, waiting for a long time to complete its execution.

Preemptive Priority Scheduling (PPS)

Using the below given Process Table, applying Preemptive Priority Scheduling Algorithm. Find the following by drawing appropriate GANTT Chart

1. Waiting Time (WT) for each process
2. Turn Around Time (TAT) for each process
3. Response Time (RT) for each process
4. Average Response Time (ART)
5. Average Waiting Time (AWT)
6. Average Turnaround Time (ATAT)

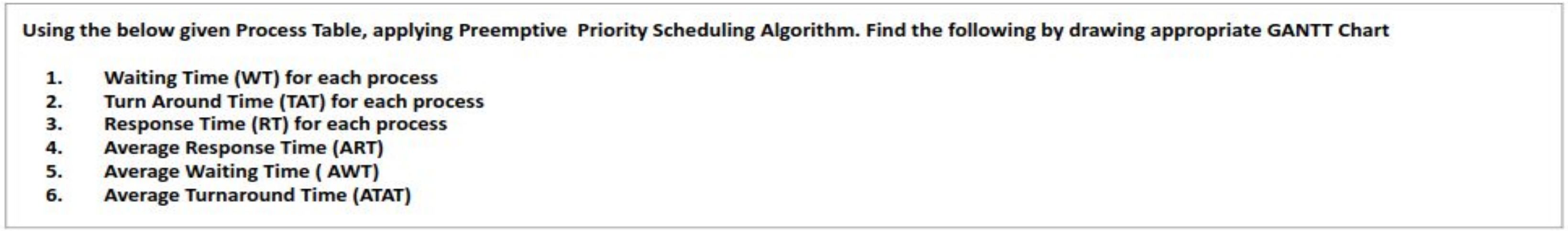
GANTT Chart



Process Table			
Process ID	Arrival Time	CPU Burst	Priority
P0	2	6	1
P1	5	2	2
P2	1	8	1
P3	0	3	1
P4	4	4	4

Process ID	Arrival Time	Response Time	Waiting Time	Turn Around time
CIP	Omnipresent	Omnipresent	Omnipresent	Omnipresent
P0	2			
P1	5			
P2	1			
P3	0			
P4	4			
Average	2.4			
Remarks	AAT	ART	AWT	ATAT

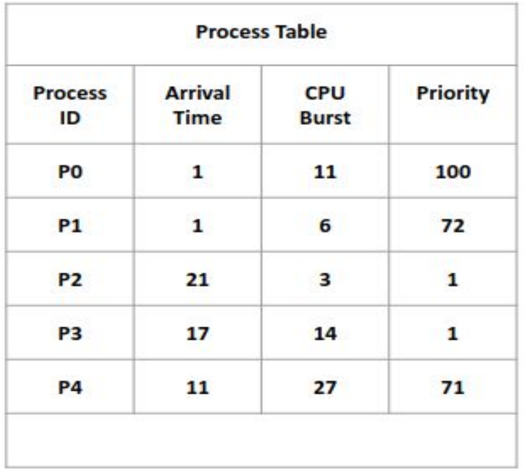
Preemptive Priority Scheduling (PPS) - Practice Problem



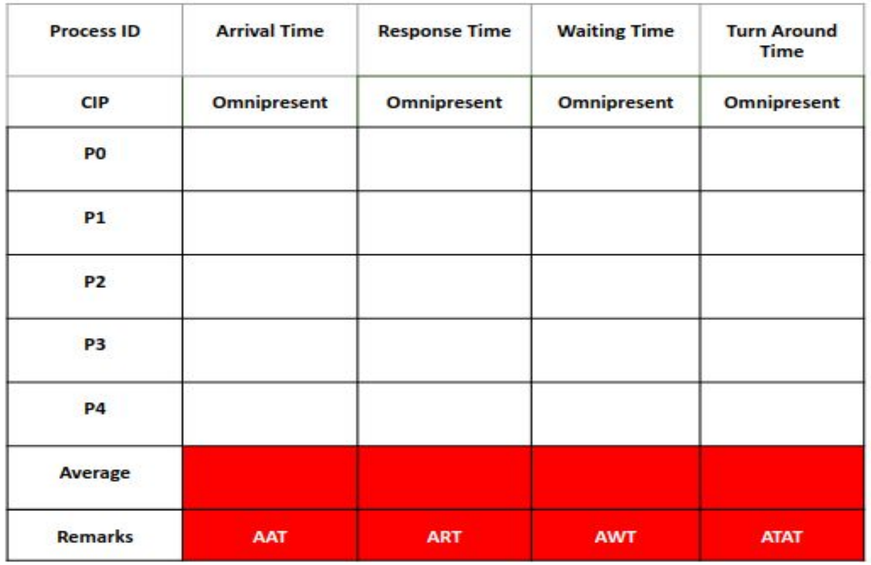
Using the below given Process Table, applying Preemptive Priority Scheduling Algorithm. Find the following by drawing appropriate GANTT Chart

1. Waiting Time (WT) for each process
2. Turn Around Time (TAT) for each process
3. Response Time (RT) for each process
4. Average Response Time (ART)
5. Average Waiting Time (AWT)
6. Average Turnaround Time (ATAT)

GANTT Chart



Process Table			
Process ID	Arrival Time	CPU Burst	Priority
P0	1	11	100
P1	1	6	72
P2	21	3	1
P3	17	14	1
P4	11	27	71



Process ID	Arrival Time	Response Time	Waiting Time	Turn Around Time
CIP	Omnipresent	Omnipresent	Omnipresent	Omnipresent
P0				
P1				
P2				
P3				
P4				
Average				
Remarks	AAT	ART	AWT	ATAT

Round Robin Scheduling (RR)

Characteristics of Round Robin Scheduling

A CPU algorithm that schedules processes based on Time Quantum.

- Round Robin is the preemptive process scheduling algorithm.
- Each process is provided a fix time to execute, it is called a **quantum**.
- Once a process is executed for a given time period, it is preempted and other process executes for a given time period.
- Context switching is used to save states of preempted processes.
- The process that is preempted is added to the end of the queue.
- Round robin is a hybrid model which is clock-driven
- Time slice should be minimum, which is assigned for a specific task that needs to be processed. However, it may differ based on OS.
- It is a real time algorithm which responds to the event within a specific time limit.
- Round robin is one of the oldest, fairest, and easiest algorithm.
- Widely used scheduling method in traditional OS.

Round Robin Scheduling (RR)

Characteristics of Round Robin Scheduling

A CPU algorithm that schedules processes based on Time Quantum.

- 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
  a. large quantum => FIFO
  b. Quantum Small => Quantum must be large with respect to context switch, otherwise overhead is too high

Round Robin Scheduling (RR)

Advantages of Round Robin Scheduling

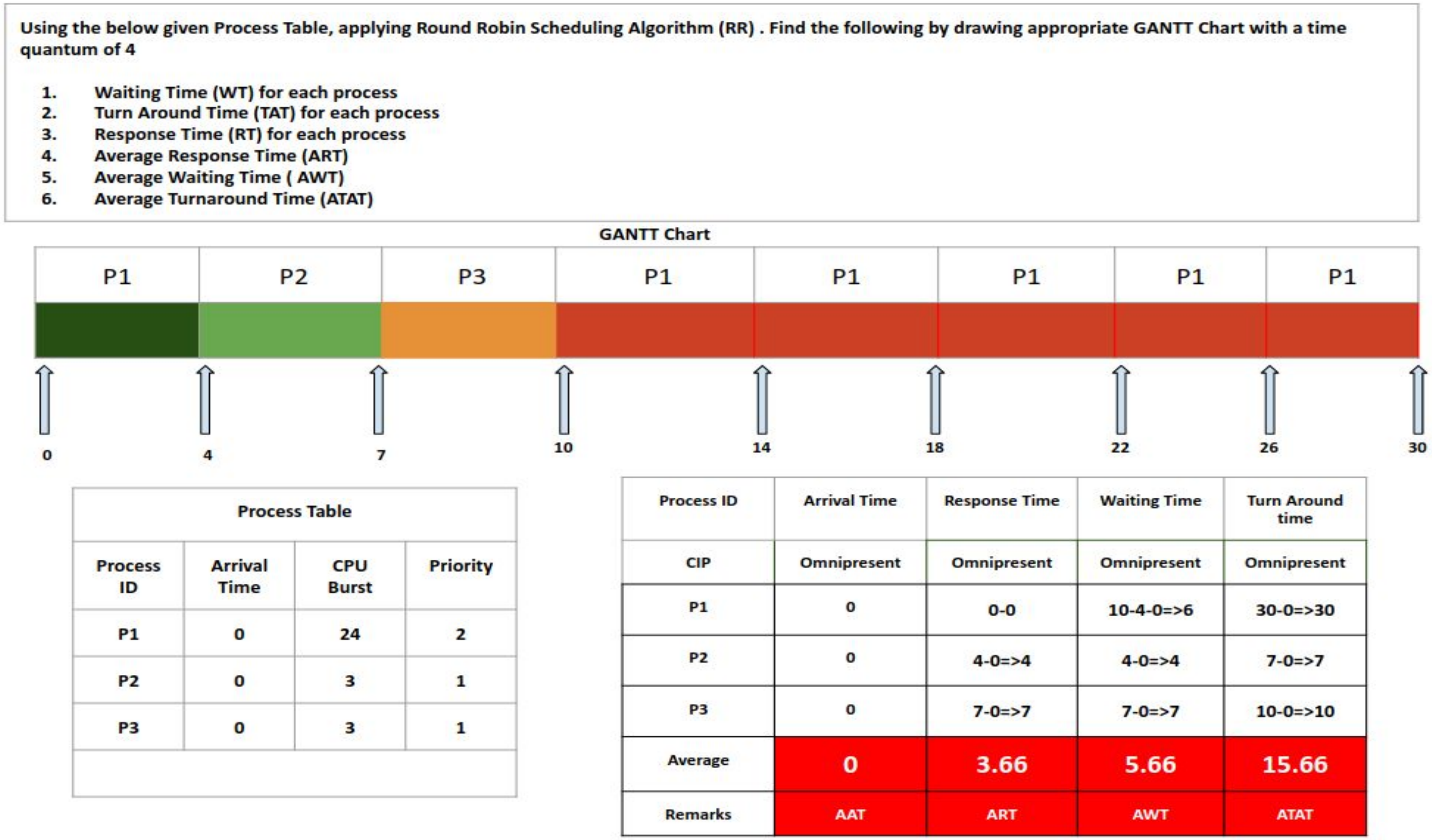
- It doesn't face the issues of starvation or convoy effect.
- All the jobs get a fair allocation of CPU.
- It deals with all process without any priority
- If you know the total number of processes on the run queue, then you can also assume the worst-case response time for the same process.
- This scheduling method does not depend upon burst time. That's why it is easily implementable on the system.
- Once a process is executed for a specific set of the period, the process is preempted, and another process executes for that given time period.
- Allows OS to use the Context switching method to save states of preempted processes.
- It gives the best performance in terms of average response time.

Disadvantages of Preemptive Priority Scheduling

- If slicing time of OS is low, the processor output will be reduced.
- This method spends more time on context switching
- Its performance heavily depends on time quantum.
- Priorities cannot be set for the processes.
- Round-robin scheduling doesn't give special priority to more important tasks.
- Decreases comprehension
- Lower time quantum results in higher the context switching overhead in the system.
- Finding a correct time quantum is a quite difficult task in this system.

OPERATING SYSTEMS

Round Robin Scheduling (RR)



Using the below given Process Table, applying Round Robin Scheduling Algorithm (RR) . Find the following by drawing appropriate GANTT Chart with a time quantum of 4

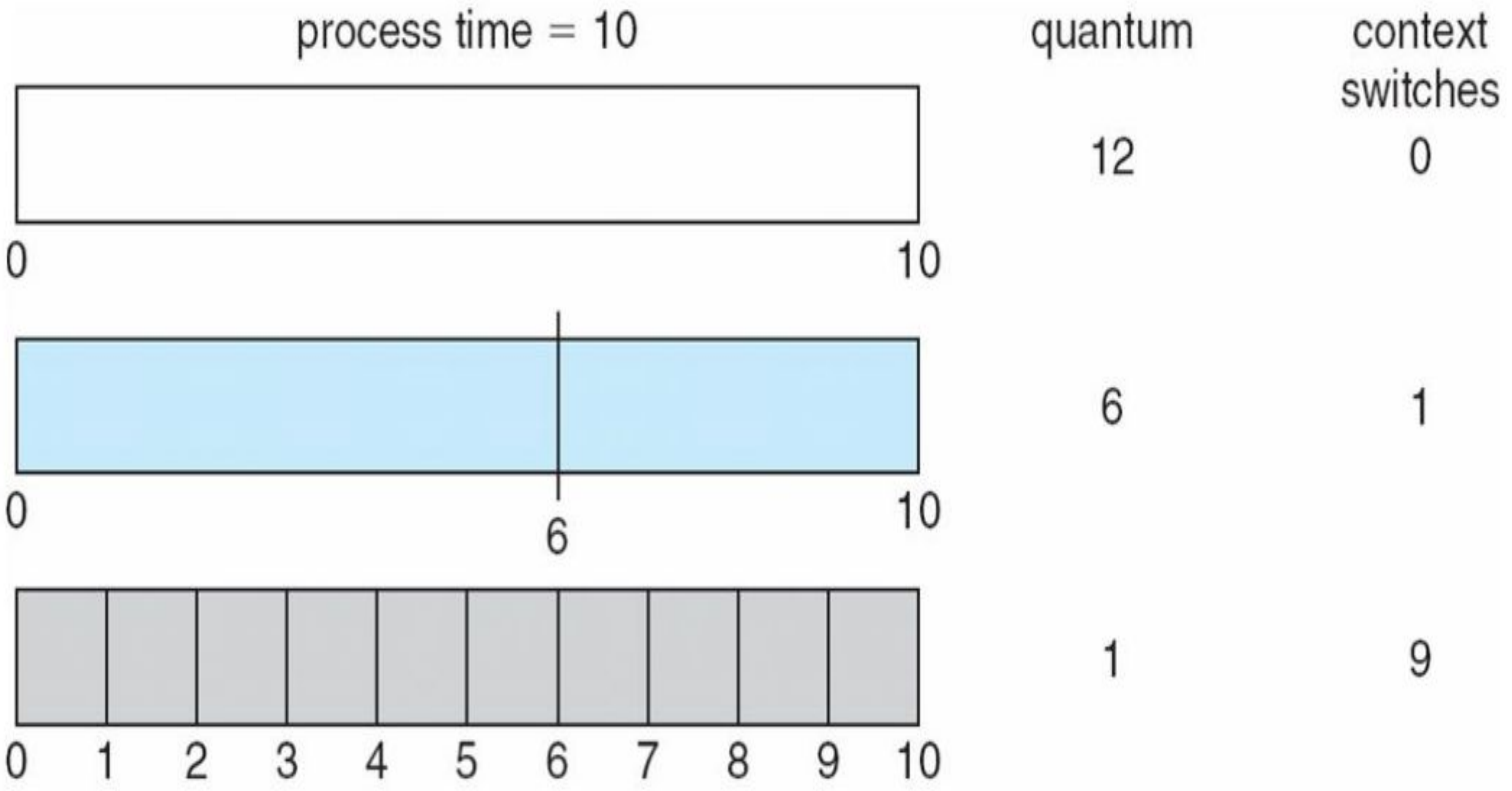
1. Waiting Time (WT) for each process
2. Turn Around Time (TAT) for each process
3. Response Time (RT) for each process
4. Average Response Time (ART)
5. Average Waiting Time (AWT)
6. Average Turnaround Time (ATAT)

GANTT Chart

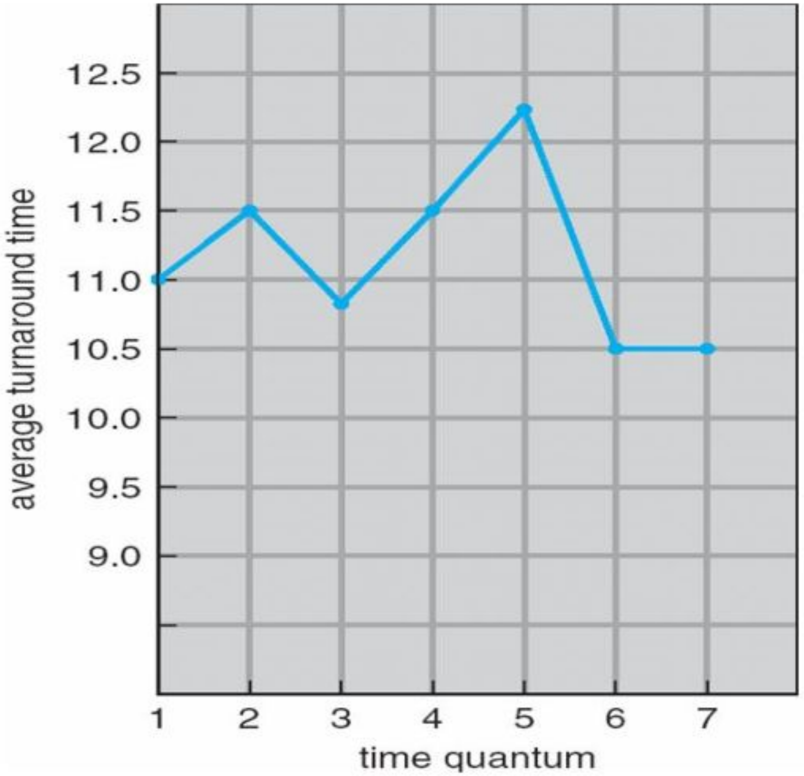
Process Table			
Process ID	Arrival Time	CPU Burst	Priority
P1	0	24	2
P2	0	3	1
P3	0	3	1

Process ID	Arrival Time	Response Time	Waiting Time	Turn Around time
CIP	Omnipresent	Omnipresent	Omnipresent	Omnipresent
P1	0	0-0	10-4-0=>6	30-0=>30
P2	0	4-0=>4	4-0=>4	7-0=>7
P3	0	7-0=>7	7-0=>7	10-0=>10
Average	0	3.66	5.66	15.66
Remarks	AAT	ART	AWT	ATAT

Round Robin Scheduling (RR) - Time Quantum and Context Switch



RR Scheduling - Time Around varies with time Quantum



process	time
P_1	6
P_2	3
P_3	1
P_4	7

80% of CPU bursts should be shorter than quantum

Round Robin Scheduling (RR) - Practice Problem

Using the below given Process Table, applying Round Robin Scheduling Algorithm (RR) . Find the following by drawing appropriate GANTT Chart with a time quantum of 12, 6, 1

1. Waiting Time (WT) for each process
2. Turn Around Time (TAT) for each process
3. Response Time (RT) for each process
4. Average Response Time (ART)
5. Average Waiting Time (AWT)
6. Average Turnaround Time (ATAT)

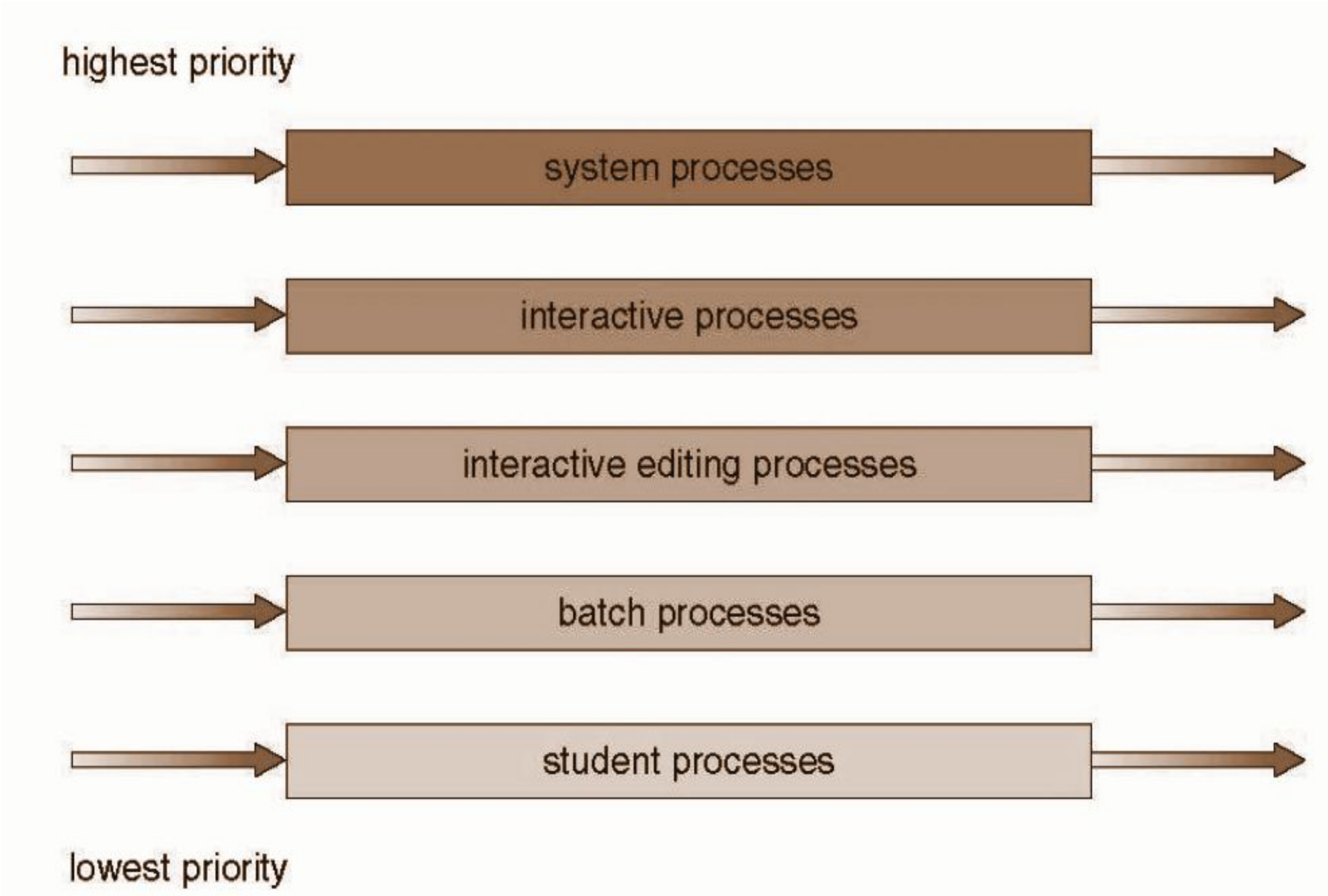
Process Table			
Process ID	Arrival Time	CPU Burst	Priority
P1	0	10	2
P2	0	3	1
P3	0	3	1

Process ID	Arrival Time	Response Time	Waiting Time	Turn Around time
CIP	Omnipresent	Omnipresent	Omnipresent	Omnipresent
P1				
P2				
P3				
Average				
Remarks	AAT	ART	AWT	ATAT

Multilevel Queues

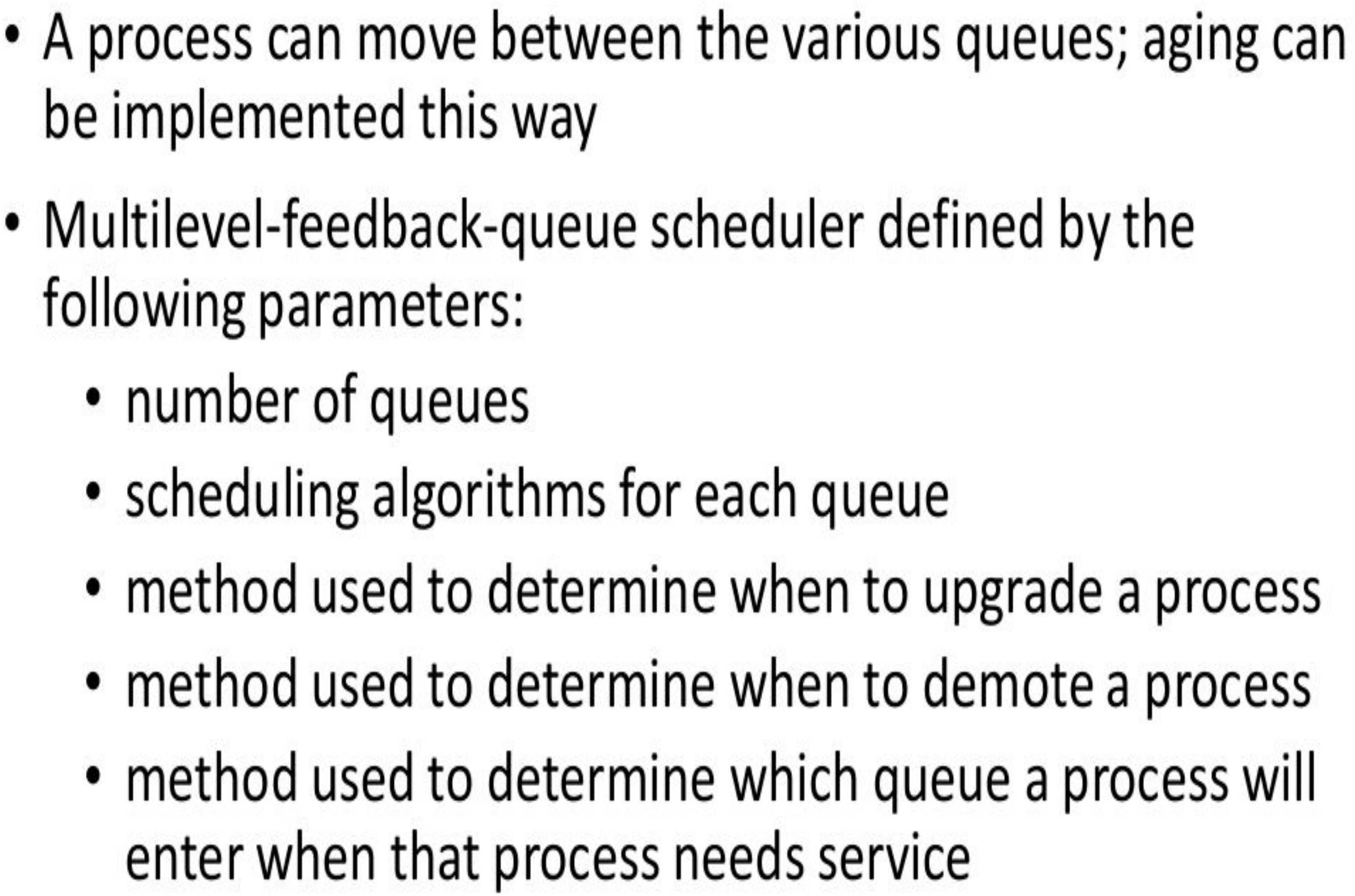
- Ready queue is partitioned into separate queues, eg:
    - **foreground** (interactive)
    - **background** (batch)
- Process permanently in a given queue
- Each queue has its own scheduling algorithm:
    - foreground – RR
    - background – FCFS
- Scheduling must be done between the queues:
    - 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

Multilevel Queues



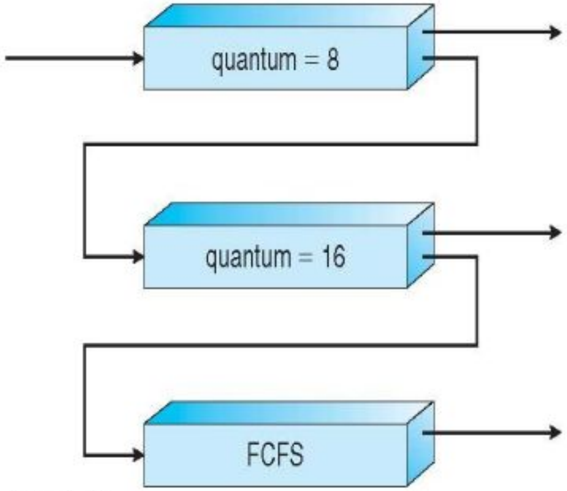
Slides Adapted from Operating System Concepts 9/e © Authors

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

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

Slides Adapted from Operating System Concepts 9/e © Authors

PES UNIVERSITY ONLINE

THANK YOU

Nitin V Pujari
Faculty, Computer Science
Dean - IQAC, PES University

nitin.pujari@pes.edu

For Course Deliverables by the Anchor Faculty click on www.pesuacademy.com