CO - 1

Session : 4

COURSE NAME : SYSTEM DESIGN AND INTRODUCTION TO CLOUD

COURSE CODE : 23AD2103A

TOPICS : CPU SCHEDULING

SESSION DESCRIPTION

- CPU SCHEDULING
- FCFS
- SJF
- SRT
- RR
- PRIORITY

Scheduling: Introduction

- The process scheduling is the activity of the process manager that handles the removal of the running process from the CPU and the selection of another process on the basis of a particular strategy.
- Workload assumptions:
  1. Each job runs for the **same amount of time.**
  2. All jobs **arrive** at the same time.
  3. All jobs only use the **CPU** (i.e., they perform no I/O).
  4. The **run-time** of each job is known.

scheduling metric:

A metric is just something that we use to measure something, and there are a number of different metrics that make sense in scheduling.

TERMINOLOGIES USED IN CPU SCHEDULING

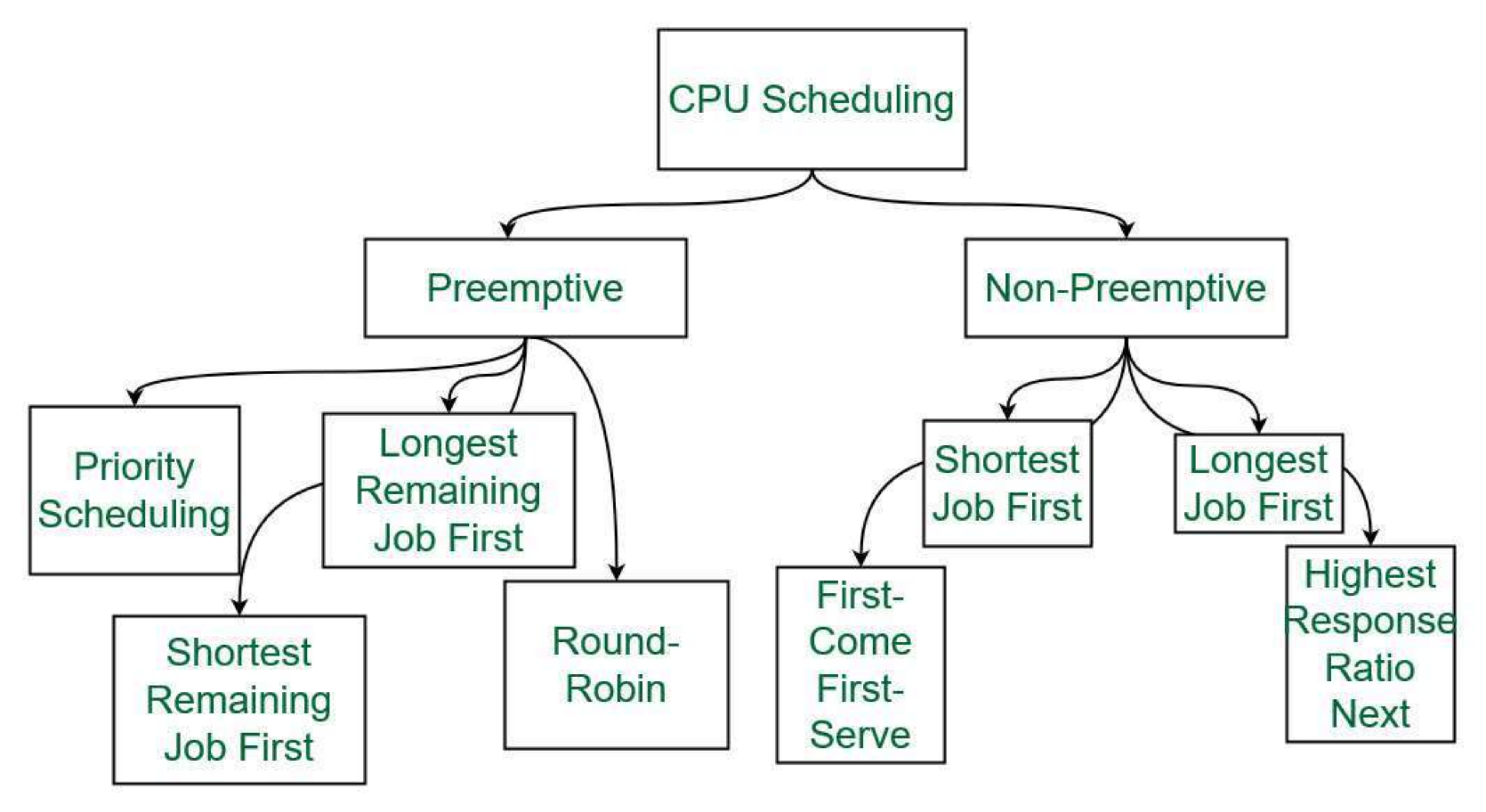
- **Arrival Time:** Time at which the process arrives in the ready queue.
- **Completion Time:** Time at which process completes its execution.
- **Burst Time:** Time required by a process for CPU execution.
- **Turn Around Time:** Time Difference between completion time and arrival time.

  →Turn Around Time = Completion Time – Arrival Time
- **Waiting Time(W.T):** Time Difference between turn around time and burst time.

  →Waiting Time = Turn Around Time – Burst Time

Preemptive and Non-Preemptive

- The basic difference between **preemptive and non-preemptive scheduling** is that:
- In **preemptive scheduling,** the CPU is allocated to the processes for the limited time.
- While in **Non-preemptive scheduling**, the CPU is allocated to the process till it terminates or switches to waiting state.



CPU Scheduling

Preemptive

Non-Preemptive

Priority Scheduling

Longest Remaining Job First

Shortest Remaining Job First

Round-Robin

Shortest Job First

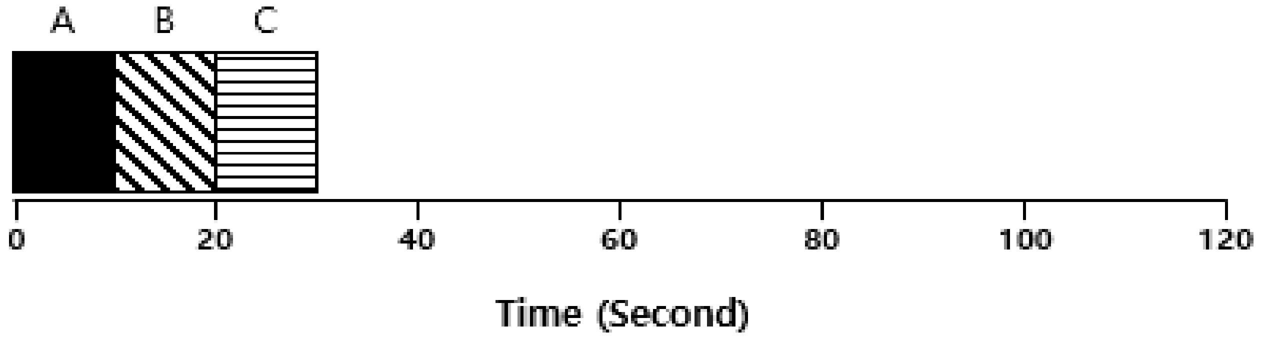
Longest Job First

First-Come First-Serve

Highest Response Ratio Next

First In, First Out (FIFO)

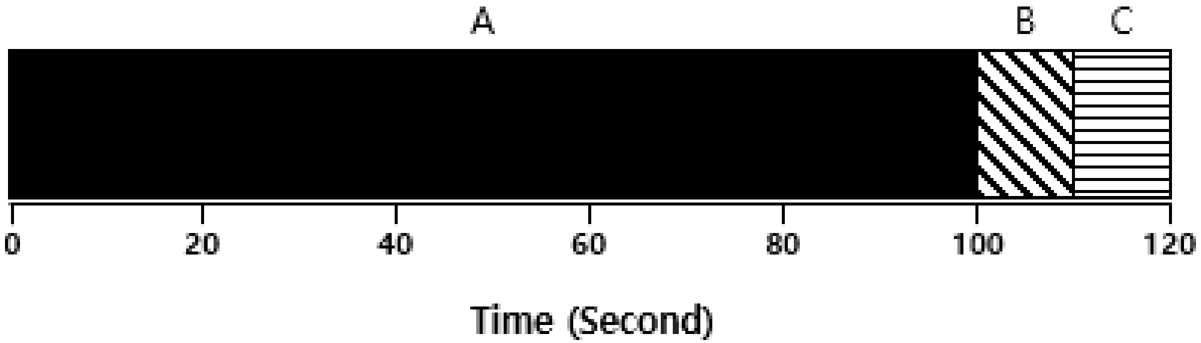
- First Come, First Served (FCFS)
  - Very simple and easy to implement
- Example:
  - A arrived just before B which arrived just before C.
  - Each job runs for 10 seconds.



$$\boldsymbol{Average\ turnaround\ time = \frac{10+20+30}{3} = 20\ sec}$$

Why FIFO is not that great? – Convoy effect

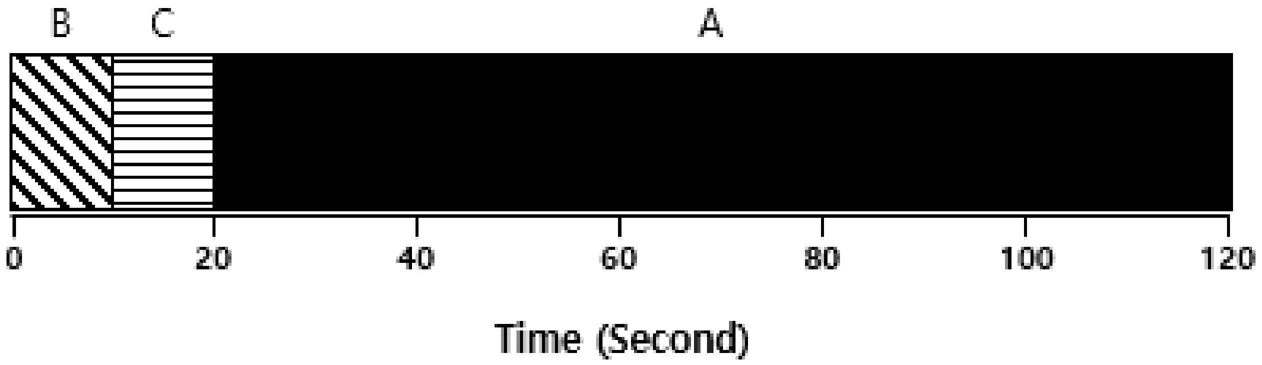
- Let’s relax assumption 1: Each job **no longer** runs for the same amount of time.
- Example:
  - A arrived just before B which arrived just before C.
  - A runs for 100 seconds, B and C run for 10 each.



$$Average\ turnaround\ time = \frac{100 + 110 + 120}{3} = 110\ sec$$

Shortest Job First (SJF)

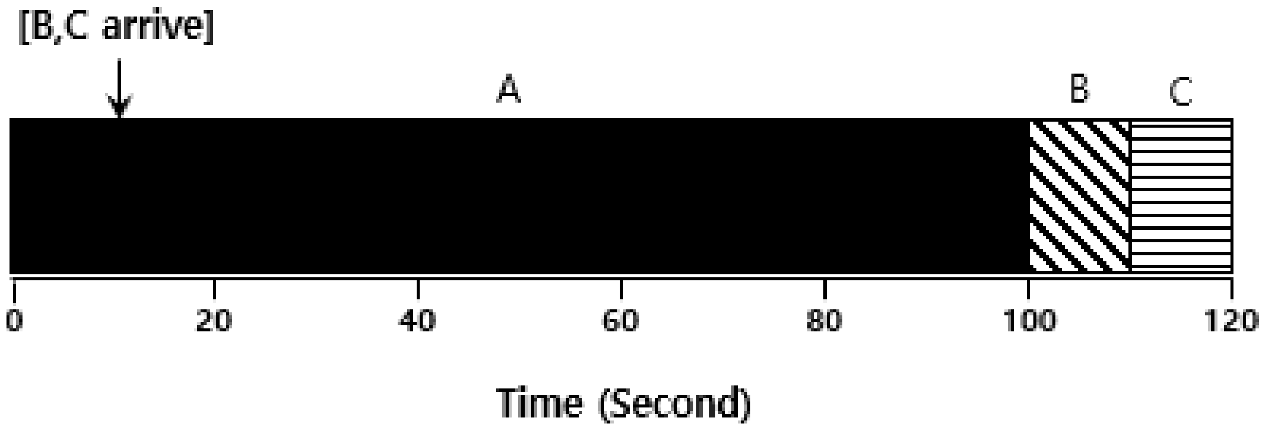
- Run the shortest job first, then the next shortest, and so on
  - Non-preemptive scheduler
- Example:
  - A arrived just before B which arrived just before C.
  - A runs for 100 seconds, B and C run for 10 each.



$$\boldsymbol{Average\ turnaround\ time = \frac{10+20+120}{3} = 50\ sec}$$

SJF with Late Arrivals from B and C

- Let’s relax assumption 2: Jobs can arrive at any time.
- Example:
  - A arrives at t=0 and needs to run for 100 seconds.
  - B and C arrive at t=10 and each need to run for 10 seconds



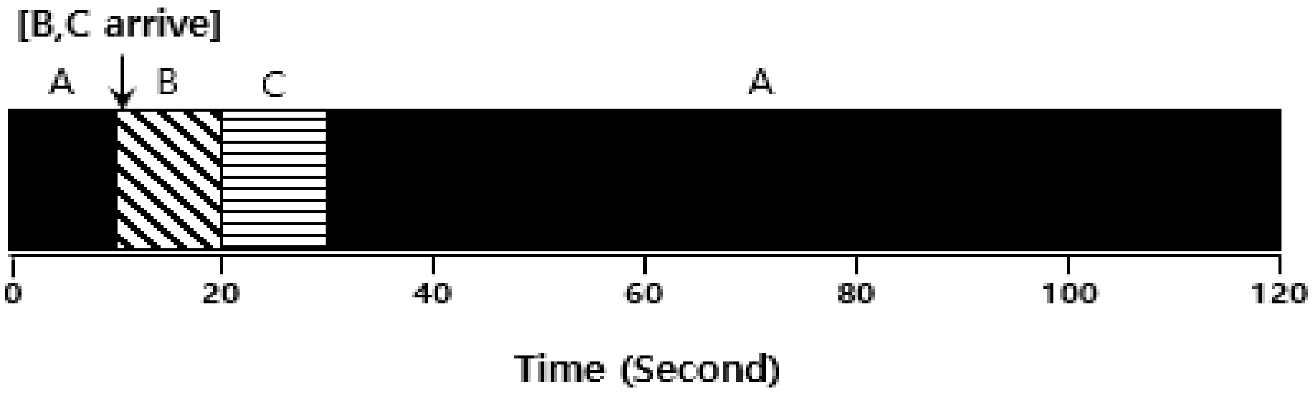
$$\boldsymbol{Average\ turnaround\ time} = \frac{100 + (110 - 10) + (120 - 10)}{3} = 103.33\ sec$$

Shortest Time-to-Completion First (STCF)

- Add preemption to SJF
  - Also knows as Preemptive Shortest Job First (PSJF)
- A new job enters the system:
  - Determine of the remaining jobs and new job
  - Schedule the job which has the lest time left

Shortest Time-to-Completion First (STCF)

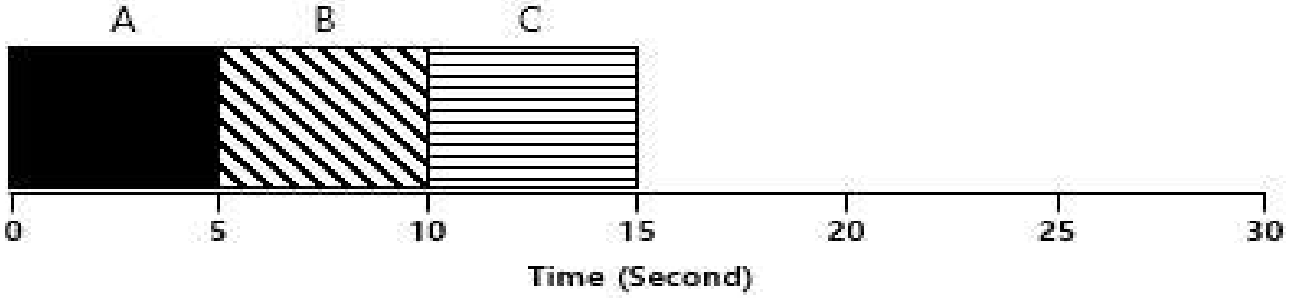
- Example:
  - A arrives at t=0 and needs to run for 100 seconds.
  - B and C arrive at t=10 and each need to run for 10 seconds



$$\textit{Average turnaround time} = \frac{(120-0)+(20-10)+(30-10)}{3} = 50\ \textit{sec}$$

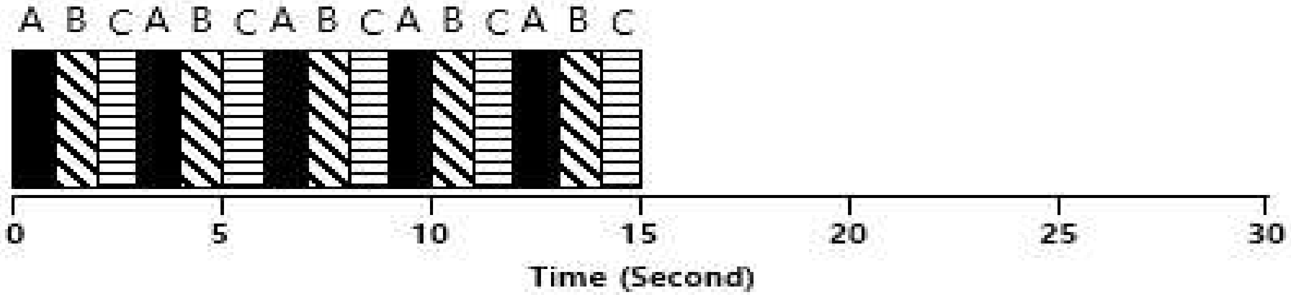
RR Scheduling Example

- A, B and C arrive at the same time.
- They each wish to run for 5 seconds.



SJF (Bad for Response Time)

$$T_{average\ response} = \frac{0+5+10}{3} = 5sec$$

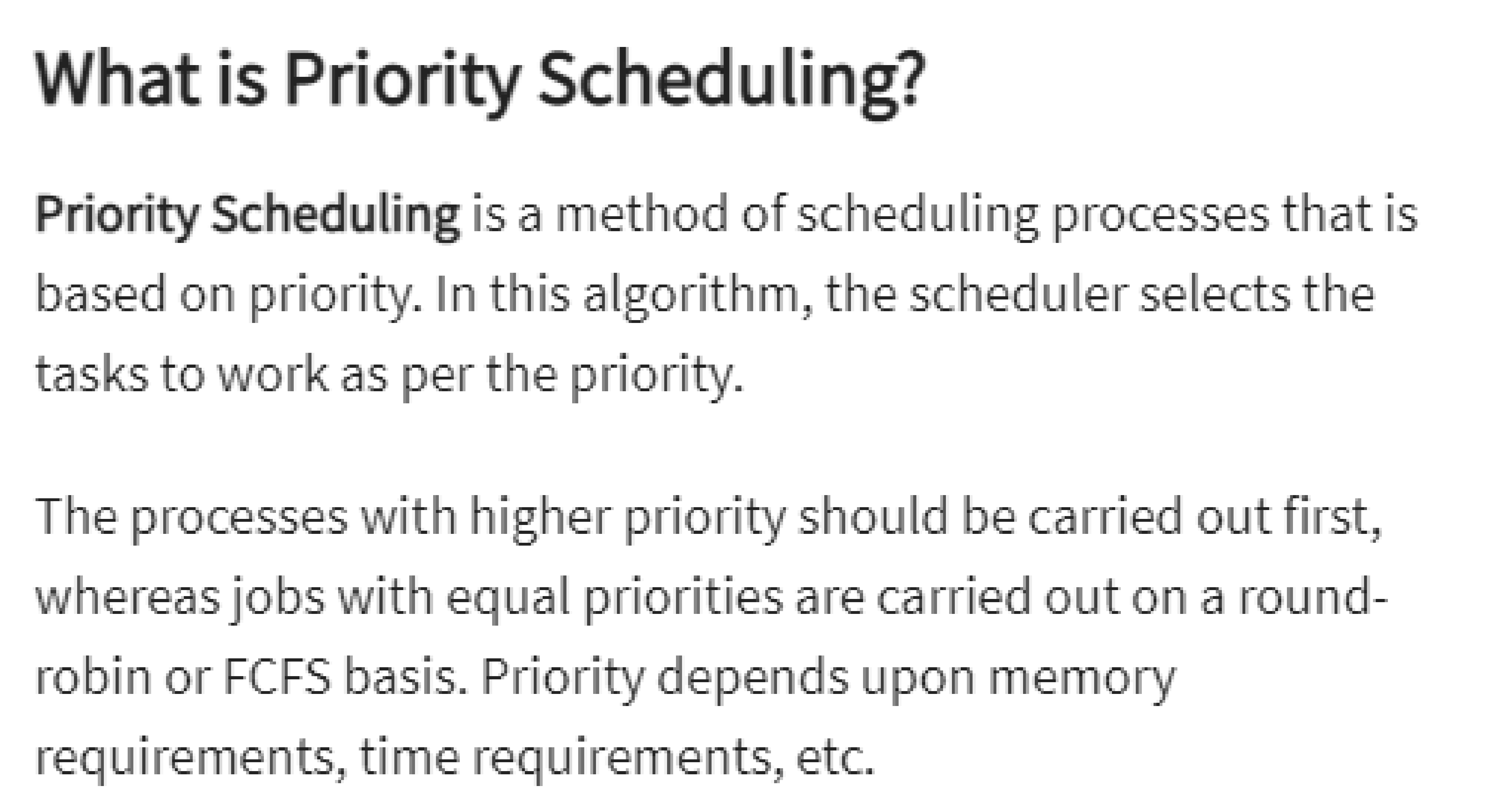


RR with a time-slice of 1sec (Good for Response Time)

$$T_{average\ response} = \frac{0+1+2}{3} = 1sec$$

THE LENGTH OF THE TIME SLICE IS CRITICAL

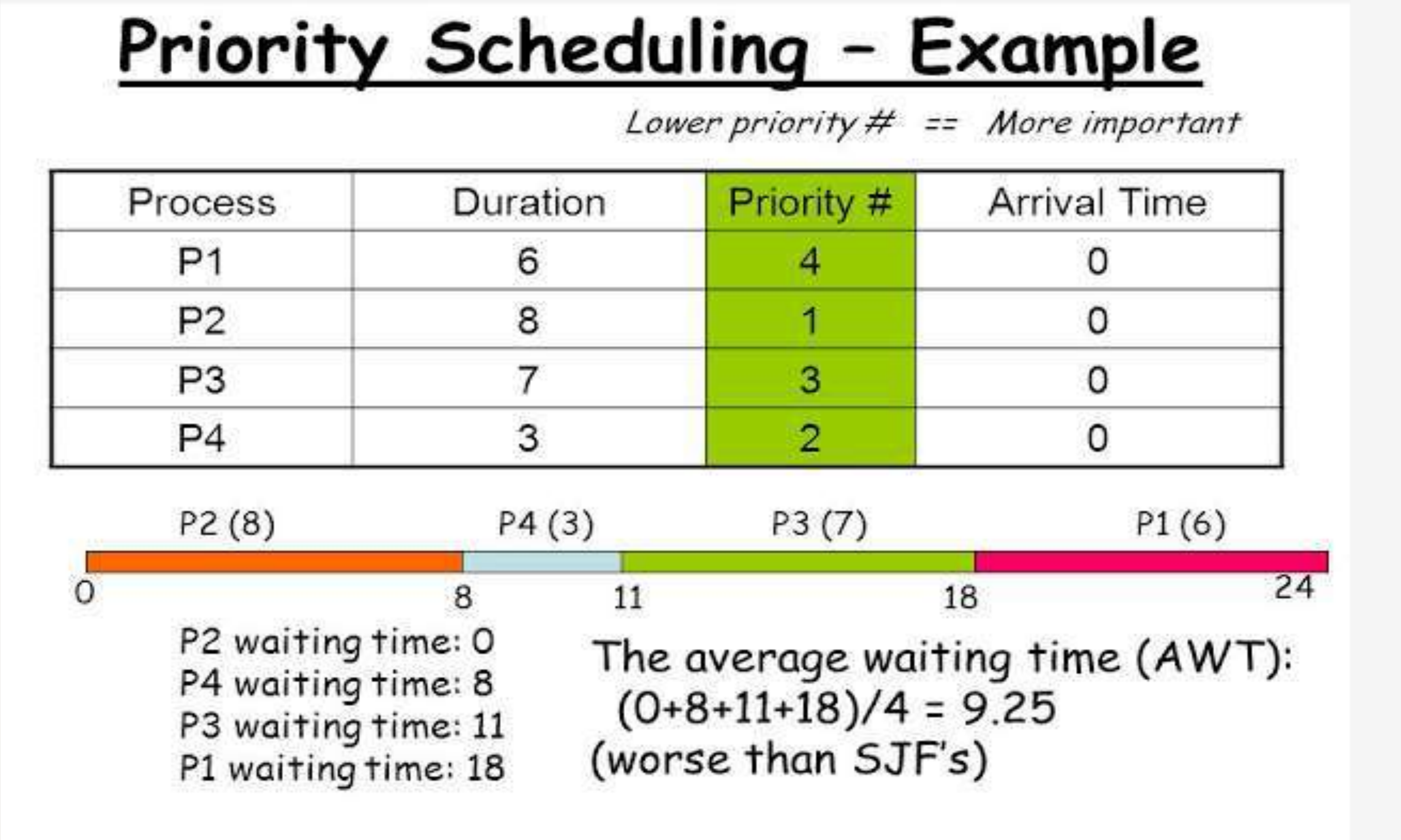
- The shorter it is, the better the performance of RR under the response time metric
- However, making the time slice too short is problematic, suddenly the cost of context switching will dominate overall performance.
- Thus, deciding on the length of the time slice presents a trade off to a system designer, making it long enough to amortize the cost of switching without making it so long that the system is no longer responsive



What is Priority Scheduling?

Priority Scheduling is a method of scheduling processes that is based on priority. In this algorithm, the scheduler selects the tasks to work as per the priority.

The processes with higher priority should be carried out first, whereas jobs with equal priorities are carried out on a round-robin or FCFS basis. Priority depends upon memory requirements, time requirements, etc.



Priority Scheduling - Example

Lower priority # == More important

Process	Duration	Priority #	Arrival Time
P1	6	4	0
P2	8	1	0
P3	7	3	0
P4	3	2	0

P2 (8)
P4 (3)
P3 (7)
P1 (6)
0
8
11
18
24

P2 waiting time: 0
P4 waiting time: 8
P3 waiting time: 11
P1 waiting time: 18

The average waiting time (AWT):
(0+8+11+18)/4 = 9.25
(worse than SJF's)

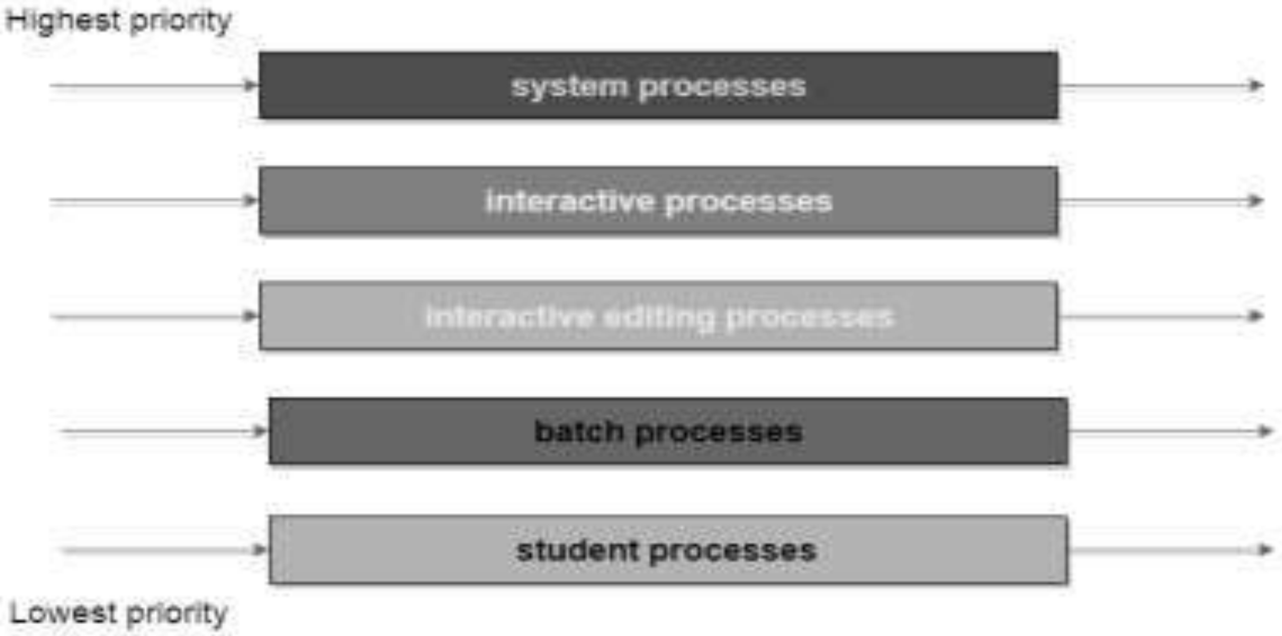
(d) Multilevel Queue Scheduling

A multi-level queue scheduling algorithm partitions the ready queue into several separate queues. The processes are permanently assigned to one queue, generally based on some property of the process, such as memory size, process priority, or process type. Each queue has its own scheduling algorithm.

Let us consider an example of a multilevel queue-scheduling algorithm with five queues:

1. System Processes
2. Interactive Processes
3. Interactive Editing Processes
4. Batch Processes
5. Student Processes

Each queue has absolute priority over lower-priority queues. No process in the batch queue, for example, could run unless the queues for system processes, interactive processes, and interactive editing processes were all empty. If an interactive editing process entered the ready queue while a batch process was running, the batch process will be pre-empted.



THANK YOU

(DEEMED TO BE UNIVERSITY)

Team – System Design & Introduction to Cloud