CPU SCHEDULING ALGORITHM

Submitted by:

ASTHA GUPTA(2100290110038)

ISHAN SHARMA(2100290110065)

HIMANSHI TYAGI(2100290110064)

What is CPU Scheduling??

- CPU scheduling is the process of determining the order in which processes or tasks are executed on the central processing unit (CPU) of a computer system.
- It involves selecting a process from the ready queue and allocating CPU time to it, based on specific scheduling algorithms and policies.
- The primary goal of CPU scheduling is to optimize CPU resource utilization, improve system performance, and provide fairness among competing processes.

Types:

Preemptive and non-preemptive are two different approaches used in CPU scheduling algorithms.

- Preemptive Scheduling: Allows a running process to be interrupted and replaced by a higher-priority process. Provides better responsiveness and ensures time-critical tasks are promptly executed. Examples include Round Robin and Priority Scheduling (dynamic priority).

- Non-Preemptive Scheduling: Once a process starts executing, it continues until completion or voluntary release. Simpler to implement but may result in delays for higher-priority processes. Examples include First-Come, First-Served and Priority Scheduling (static priority).

Scheduling Algorithms

- **First-Come, First-Served (FCFS):**
- **Shortest Job First (SJF)**
- **Round Robin (RR):**
- **Priority Scheduling:**

Process Scheduling Solution

Process	Burst Time
P1	9
P2	33
P3	2
P4	5
P5	14

Suppose the process arrive in the order P1,P2,P3,P4,P5

FCFS(First-Come, First-Served)

For FCFS the process would be executed in the following order, with the following wait times

Process	Burst Time	Wait Time
P1	9	0
P2	33	9
P3	2	42
P4	5	44
P5	14	49

Therefore, the average waiting time is ((0 + 9 + 42 + 44 + 49) / 5) = 28.80 milliseconds

SJF(Shortest Job First)

For SJF (non preempted) the processes would run as follows

Process	Burst Time	Wait Time
P3	2	0
P4	5	2
P1	9	7
P5	14	16
P2	33	30

The average waiting time is ((0 + 2 + 7 + 16 + 30) / 5) = 11 milliseconds

Priority Scheduling

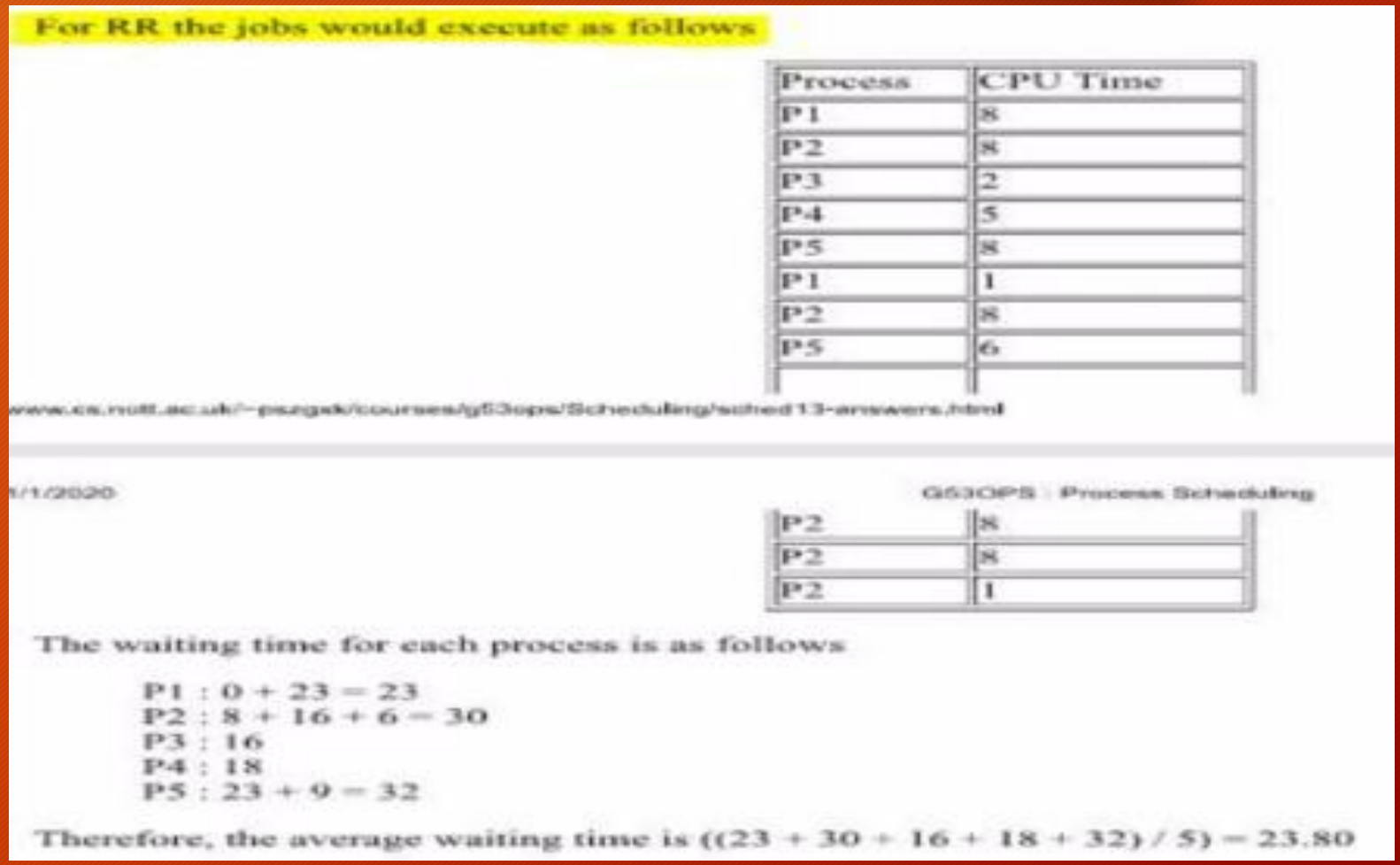
Process	Burst Time	Priority
P_1	10	3
P_2	1	1
P_3	2	4
P_4	1	5
P_5	5	2

- Priority scheduling Gantt Chart

- Average waiting time = 8.2 msec

RR (Round Robin)

- Let the quantum is 8



For RR the jobs would execute as follows

Process	CPU Time
P1	8
P2	8
P3	2
P4	5
P5	8
P1	1
P2	8
P5	6

www.cs.nott.ac.uk/~pszgsk/courses/g53ops/Scheduling/sched13-answers.html

1/1/2020 G53OPS : Process Scheduling

P2	8
P2	8
P2	1

The waiting time for each process is as follows

P1 : 0 + 23 = 23
P2 : 8 + 16 + 6 = 30
P3 : 16
P4 : 18
P5 : 23 + 9 = 32

Therefore, the average waiting time is ((23 + 30 + 16 + 18 + 32) / 5) = 23.80

Multiprocessor Scheduling

Multiprocessor scheduling refers to the allocation of processes across multiple processors in a parallel computing system. It involves efficiently distributing the workload among processors to maximize overall system performance and resource utilization. The goal is to balance the load, minimize idle time, and ensure fair and efficient execution of processes across the available processors. Various scheduling algorithms and techniques are used to achieve effective multiprocessor scheduling in order to exploit the benefits of parallel processing.

Real-Time Scheduling

- Real-time scheduling is a specific type of scheduling used in real-time systems, where meeting strict timing constraints is critical. Here are three key points about real-time scheduling:

- Timing Guarantees: Real-time scheduling ensures that tasks or processes are completed within specific time constraints, known as deadlines. These deadlines can be hard (must be met) or soft (preferred but noted mandatory). Meeting these timing guarantees is essential for the correct operation of real-time systems.

- Priority-based Scheduling: Real-time scheduling often employs priority-based algorithms, where tasks are assigned priorities based on their timing requirements. Higher priority tasks are given precedence over lower priority tasks. This ensures that time-critical tasks receive immediate attention and are executed within their deadlines.

References

- https://www.geeksforgeeks.org/cpu-scheduling-in-operating-systems/
- https://youtu.be/EWkQl0n0w5M

GitHub Link:

https://github.com/asthaisback/CPU-Scheduling


THANK YOU!