Scheduling Algorithms: Problems and Solutions

Rate Monotonic Scheduling (RMS)

Problem:

Consider the set of 5 processes whose details are given below:

Process Id	Arrival Time	Burst Time	Period
P1	0	1	4
P2	0	2	6
P3	0	1	8

If the CPU scheduling policy is **Rate Monotonic Scheduling**, calculate the CPU utilization and determine if the tasks are schedulable.

Solution:

Rate Monotonic Scheduling prioritizes tasks with shorter periods. CPU utilization for RMS is given by:

$$U = \sum_{i=1}^{n} \frac{\text{Burst Time}_i}{\text{Period}_i}$$

- For P1: $\frac{1}{4} = 0.25$
- For P2: $\frac{2}{6} = 0.333$
- For P3: $\frac{1}{8} = 0.125$

$$U = 0.25 + 0.333 + 0.125 = 0.708$$

The RMS bound for 3 processes is:

$$U_{\max} = n(2^{1/n} - 1) \approx 0.779$$

Since $U = 0.708 < 0.779$, the tasks are schedulable.

Earliest Deadline First (EDF)

Problem:

Consider the same set of processes with their deadlines equal to their periods. Using **Earliest Deadline First** scheduling, create a schedule for the first 10 units of time.

Solution:

EDF dynamically assigns priorities based on deadlines. The process with the earliest deadline gets the highest priority.

Gantt Chart:

Time	Executing Process
0-1	P1
1-3	P2
3-4	P1
4-5	P3
5-6	P1
6-8	P2
8-9	P1
9-10	P3

Analysis:

- **Utilization** is the same as RMS: 0.708.
- EDF guarantees scheduling since $U \leq 1$.

Proportional Share Scheduling

Problem:

Consider the same set of processes. Allocate a proportional share of CPU time to each process based on their weight. Assume the weights are as follows:

Process Id	Weight
P1	3
P2	2
P3	1

Solution:

Proportional Share Scheduling ensures that each process receives CPU time proportional to its weight.

Proportional Share Scheduling ensures that each process receives CPU time proportional to its weight.

Time Allocation for 10 Units:

- Total weight: $3 + 2 + 1 = 6$
- Share for P1: $\frac{3}{6} \times 10 = 5$ units
- Share for P2: $\frac{2}{6} \times 10 \approx 3.33$ units
- Share for P3: $\frac{1}{6} \times 10 \approx 1.67$ units

Gantt Chart:

Time	Executing Process
0-5	P1
5-8	P2
8-10	P3

Analysis:
Proportional share scheduling ensures fair dis↓ution of CPU time based on weights.

Given three tasks with the following characteristics:

- Task A: Period = 4, Execution Time = 1, Deadline = 4
- Task B: Period = 6, Execution Time = 2, Deadline = 6
- Task C: Period = 8, Execution Time = 2, Deadline = 8

Compute the Processor utilization and show the scheduling using Earliest Deadline First Scheduling algorithm for first 12 units

Processor Utilization is calculated as:

$$U = \sum_{i=1}^{n} \frac{\text{Execution Time}_i}{\text{Period}_i}$$

Substituting the given values:

- For Task A: $\frac{1}{4} = 0.25$
- For Task B: $\frac{2}{6} \approx 0.333$
- For Task C: $\frac{2}{8} = 0.25$

$$U = 0.25 + 0.333 + 0.25 = 0.833$$

The processor utilization is **83.3%**. Since this is less than **100%**, the system is schedulable under EDF.

Step 2: Schedule Tasks Using EDF

EDF Scheduling Rule: At any time, the task with the earliest absolute deadline is given the CPU.

Task Characteristics and Instances

- **Task A:** Period = 4, Execution Time = 1, Deadlines = 4, 8, 12...
- **Task B:** Period = 6, Execution Time = 2, Deadlines = 6, 12...
- **Task C:** Period = 8, Execution Time = 2, Deadlines = 8, 16...

Gantt Chart Construction

We evaluate tasks for execution based on their deadlines:

Time	Executing Task	Deadline of Task
0-1	A	4
1-3	B	6
3-4	A	8
4-6	C	8
6-7	A	8
7-9	B	12
9-10	A	12
10-12	C	16

Analysis of Task Schedules

- **Task A:** Instances scheduled at 0, 3, 6, and 9 (Execution Time = 1 unit each).
- **Task B:** Instances scheduled at 1 and 7 (Execution Time = 2 units each).
- **Task C:** Instances scheduled at 4 and 10 (Execution Time = 2 units each).