Real-Time CPU Scheduling Algorithms:

The precise scheduling algorithms are necessary for real-time systems in order to guarantee that the tasks are completed by the due date without any error. Among the various methods, Least Laxity First (LLF), Deadline Monotonic Scheduling (DMS), and Minimal Slack Scheduling (MSS) are the most popular. Here I am giving a summary report that examines their logic, working principles, advantages, and limitations, followed by a comparative analysis.

1. Deadline Monotonic Scheduling (DMS):

Deadline monotonic priority assignment is a priority assignment policy used with fixed priority preemptive scheduling.

Working Principle:

The DMS is the strict planner who sorts tasks by urgency. If a task has a deadline in 5 minutes, it jumps to the top of the list. It prioritizes deadlines over the listed number of the assignments. Whichever task is close to its deadline that will be prioritized.

Advantages:

- The tasks are prioritized by their deadlines. Shorter deadline, Higher Priority.
- The currently running tasks will preempt if a higher priority task comes.

Limitations:

- It is not optimal for dynamic tasks.
- It may suffer from priority inversions if it is not combined with the protocols like Priority Inheritance.

---

2. Least Laxity First (LLF):

Least Laxity First (LLF) is a job level dynamic priority scheduling algorithm.The main goal of the LLF algorithm is to minimize the laxity (the amount of time a task can be delayed without missing its deadline) of each task.

Working Principle:

It computes the laxity of each task and identifies the smallest laxity value. The task with the least laxity has the highest priority because it is the closest task to miss its deadline. Then it continues this process, selecting and executing tasks with the least laxity until all tasks have been completed.

Advantages:

- It is highly responsive for dynamic working schedules.
- It is optimal for meeting deadlines when feasible and efficiently handles sporadic and aperiodic tasks.

Limitations:

- There is a high runtime overhead due to the continuous laxity updates.
- It is complex to implement in practice and may lead to frequent context switches, increasing system loads.

---

3. Minimal Slack Scheduling (MSS):

Minimal Slack Scheduling (MSS) refers to a project scheduling technique where tasks are scheduled to use the minimum amount of slack time, also known as float.

Working Principle:

It computes the slack time for each task and then the scheduler picks the task with the smallest slack. Here also the preemptive occurs if a new task has less slack than the current one.
The slack is compute with this formula,

$$\text{Slack} = \text{Deadline} - \text{Remaining Execution Time}$$

Advantages:

- It is effective for dynamic environments.
- It ensures critical tasks are executed before the deadlines and it is adaptable to varying workloads.

Limitations:

- There is a high computational overhead due to slack recalculation.

- There is a risk of excessive preemptions that leads to inefficiency. Moreover, it is not always optimal for large workloads.

---

2. Comparative Analysis:

Criteria	DMS	LLF	MSS
Priority Type	Fixed	Dynamic	Dynamic
Basis of Scheduling	Deadline (shorter = higher)	Laxity (least first)	Slack (minimal first)
Overhead	Low	High	High
Preemption	Yes (priority-based)	Yes (laxity-based)	Yes (slack-based)
Optimality	Optimal for static priorities	Optimal for dynamic cases	Near-optimal for dynamic cases
Use Case	Periodic real-time systems	Dynamic real-time systems	Mixed real-time workloads
Complexity	Simple	Complex	Complex

3. Conclusion:

Each algorithm has its strengths:

- DMS excels in simplicity and predictability for fixed workloads.
- LLF is ideal for highly dynamic systems where deadlines are critical.
- MSS offers a balance but requires careful tuning to avoid inefficiency.

The choice depends on the system's nature—static workloads favor DMS, while dynamic systems benefit from LLF or MSS, provided the overhead is manageable.