Real time task scheduling

Outline

- **Real time system**
- **Scheduling periodic tasks**
- **Real time Scheduling algorithms**
- **references**

Real Time System

- Real time systems have been defined as: “those systems in which the correctness of the system depends not only on the logical result of the computation, but also on the time at which the results are produced”
- Correct function at correct time
- Usually embedded
- Types
  - Hard real-time systems
  - Soft real-time systems

Real Time System

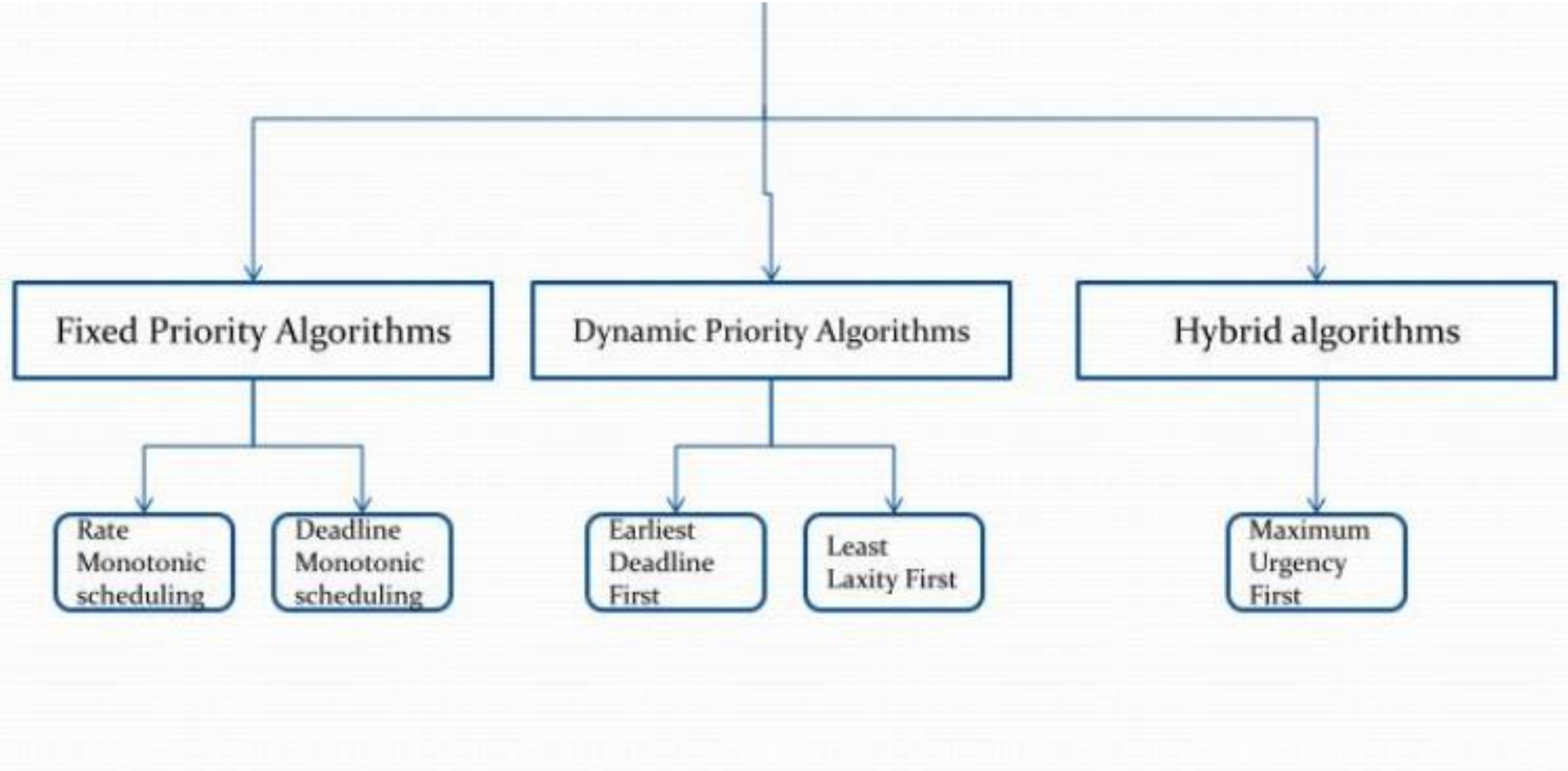
- Soft real-time systems: meet timing constraints most of the time, it is not necessary that every time constraint be met. Some deadline miss is tolerated

- Hard real-time systems: meet all timeconstraints exactly, every resource management system must work in the correct order to meet time constraints. No deadline miss is allowed.

Task categories

- Invocation
  - Periodic (time - triggered)
  - Aperiodic (event - triggered)
- Creation
  - Static
  - Dynamic
- Multi-Tasking system
  - Preemptive: higher priority process taking control of the processor from a lower priority
  - Non-Preemptive: each task can control the CPU for as long as it needs it

Scheduling algorithms



Rate monotonic scheduling

- Priority assignment based on rates of tasks
- Higher rate task assigned higher priority
- Schedulable utilization U_r

$$U = \sum_{i=1}^{n} \frac{C_i}{T_i} \leq n(\sqrt[n]{2} - 1)$$

- if $U > U_r$ schedulability is guaranteed
- Tasks may be schedulable even if $U < U_r$

Rate monotonic sheduling example

Process	Execution Time	Period
P_1	1	8
P_2	2	5
P_3	2	10

The utilization will be:

$$\frac{1}{8} + \frac{2}{5} + \frac{2}{10} = 0.725$$

The theoretical limit for processes, under which we can conclude that the system is schedulable is:

$$U = 3(2^{\frac{1}{3}} - 1) = 0.77976\ldots$$

Since $0.725 < 0.77976\ldots$ the system is schedulable!

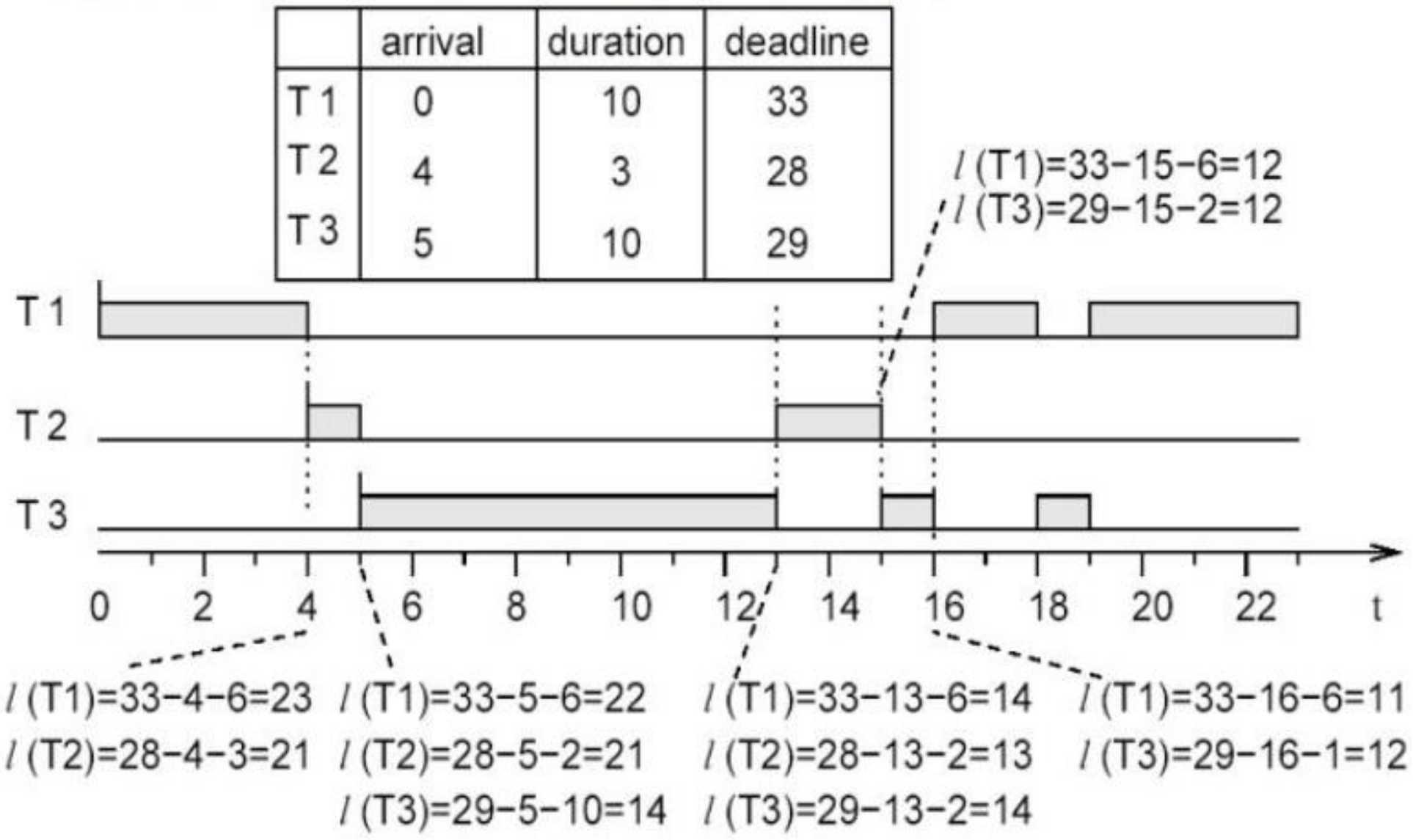
Earliset deadline first

- Dynamic priority scheduling
- Priorities are assigned according to deadlines:
  - earlier deadline, higher priority
  - later deadline, lower priority
- The first and the most effectively widely used dynamic priority-driven scheduling algorithm
- Effective for both preemptive and non-preemptive scheduling

Least Laxity First Cont

- LLF considers the execution time of task, which EDF does not
- LLF assigns higher priority to a task with the least laxity
- A task with zero laxity must be scheduled right away and executed without preemption or it will fail to meet its deadline
- The negative laxity indicates that the task will miss the deadline, no matter when it is picked up for execution

Least Laxity First Example



References

- https://en.wikipedia.org/wiki/Real-time_operating_system#Scheduling
- https://en.wikipedia.org/wiki/Rate-monotonic_scheduling
- https://barrgroup.com/Publications/Glossary/RMA.php