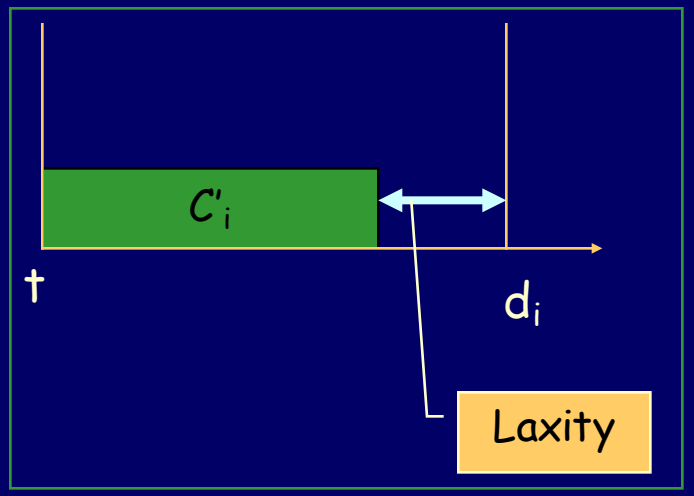
Real-Time Systems

RMS and EDF Schedulers

Priority-driven Preemptive Scheduling

Assumptions & Definitions

- Tasks are periodic
- No aperiodic or sporadic tasks
- Job (instance) deadline = end of period
- No resource constraints
- Tasks are preemptable



- Laxity of a Task
- $Ti = d_i - (t + c_i')$

  where di: deadline;

  t : current time;

  c_i' : remaining computation time.

Rate Monotonic Scheduling (RMS)

- **Schedulability check (off-line)**

  - A set of ***n*** tasks is schedulable on a uniprocessor by the RMS algorithm if the processor utilization (utilization test):

$$\sum_{i=1}^{n} c_i/p_i \leq n(2^{1/n} - 1).$$

  The term $n(2^{1/n} - 1)$ approaches $ln\ 2$, (≈ 0.69 as $n \rightarrow \infty$).

  - This condition is sufficient, but not necessary.

RMS (cont.)

- **Schedule construction (online)**

  - Task with the smallest period is assigned the highest priority.

  - At any time, the highest priority task is executed.

RMS is an optimal preemptive scheduling algorithm with fixed priorities.

Static/fixed priority algorithm assigns the same priority to all the jobs (instances) in each task.

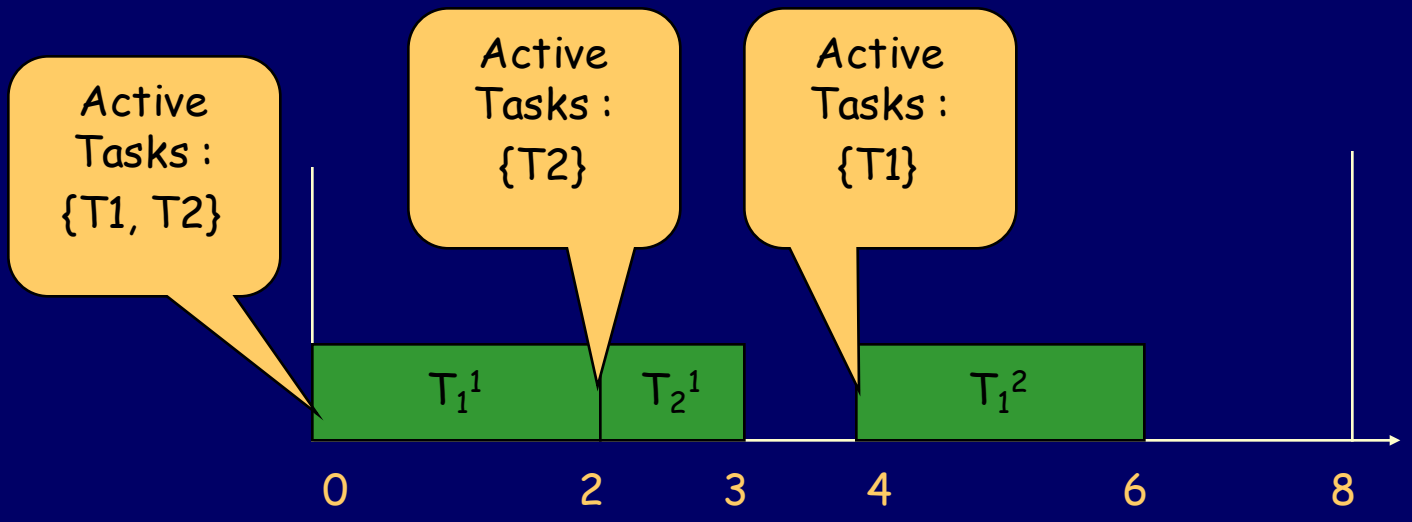
RMS Scheduler -- Example 1

Task set: $T_i = (c_i, p_i)$

T1 = (2,4) and T2 = (1,8)

Schedulability check:

$2/4 + 1/8 = 0.5 + 0.125 = 0.625 \leq 2(\sqrt{2} - 1) = 0.82$



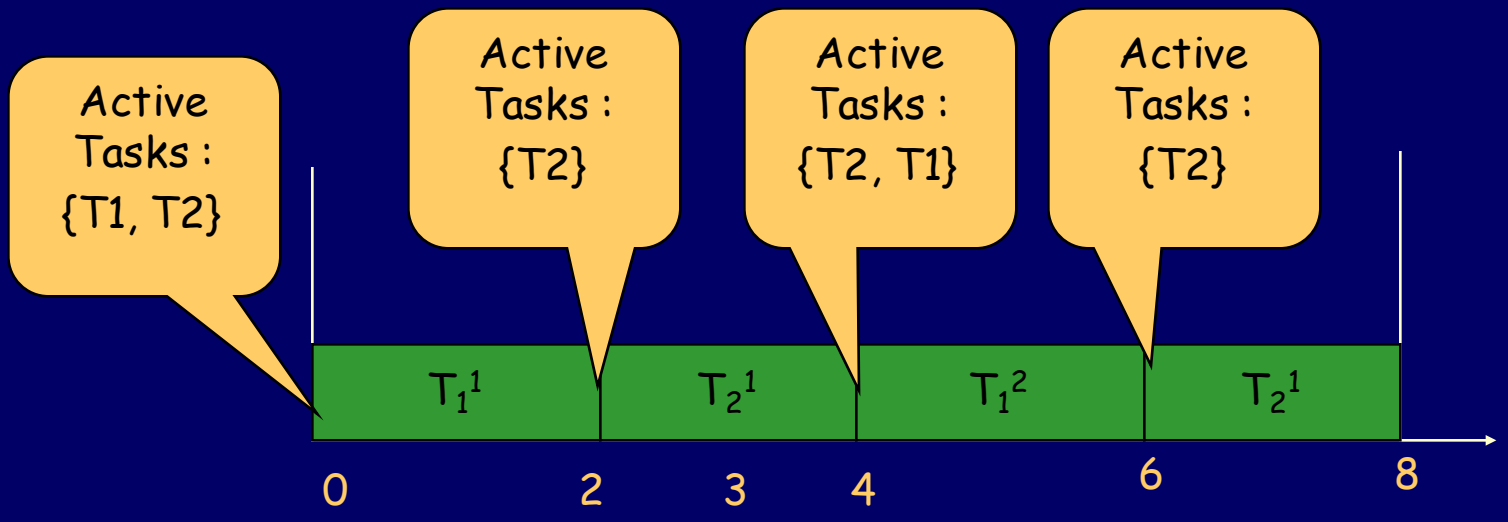
RMS scheduler -- Example-2

Task set: $T_i = (c_i, p_i)$

T1 = (2,4) and T2 = (4,8)

Schedulability check:

$2/4 + 4/8 = 0.5 + 0.5 = 1.0 > 2(\sqrt{2} - 1) = 0.82$



Some task sets that FAIL the utilization-based schedulability test are also schedulable under RMS ➔ We need exact analysis (necessary & sufficient)

Earliest Deadline First (EDF)

- **Schedulability check (off-line)**

  - A set of ***n*** tasks is schedulable on a uniprocessor by the EDF algorithm if the processor utilization.

$$\sum_{i=1}^{n} c_i / p_i \leq 1$$

- This condition is both necessary and sufficient.

- **Least Laxity First (LLF)** algorithm has the same schedulability check.

EDF/LLF (cont.)

- **Schedule construction (online)**
  - EDF/LLF: Task with the smallest deadline/laxity is assigned the highest priority.
  - At any time, the highest priority task is executed.

EDF/LLF is an optimal preemptive scheduling algorithm with dynamic priorities.

Dynamic priority algorithm assigns different priorities to the individual jobs (instances) in each task.

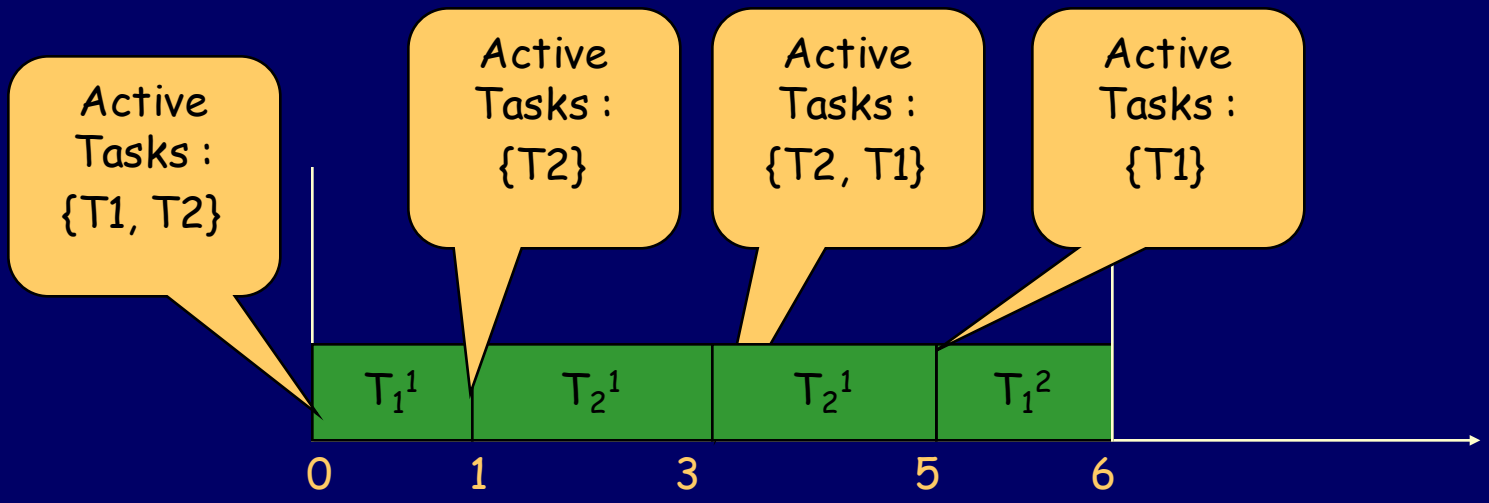
EDF scheduler -- Example

Task set: $T_i = (c_i, p_i, d_i)$

T1 = (1,3,3) and T2 = (4,6,6)

Schedulability check:

1/3 + 4/6 = 0.33 + 0.67 = 1.0



Unlike RMS, Only those task sets which pass the schedulability test are schedulable under EDF

RMS vs. EDF/LLF

- RMS is an optimal preemptive scheduling algorithm with fixed priorities.
- EDF/LLF is an optimal preemptive scheduling algorithm with dynamic priorities.
- RMS schedulability properties can be analyzed; rich theory exists and it is widely used in practice.
- EDF/LLF offers higher schedulability than RMS, but it is more difficult to implement.

RMS vs. EDF/LLF

	RMS	EDF
Priorities	Static	Dynamic
Works with OS with fixed priorities	**Yes**	**No***
Uses full computational power of processor	No, just up till $\mu = n(2^{1/n}-1)$	Yes
Possible to exploit full computational power of processor without provisioning for slack	**No**	**Yes**