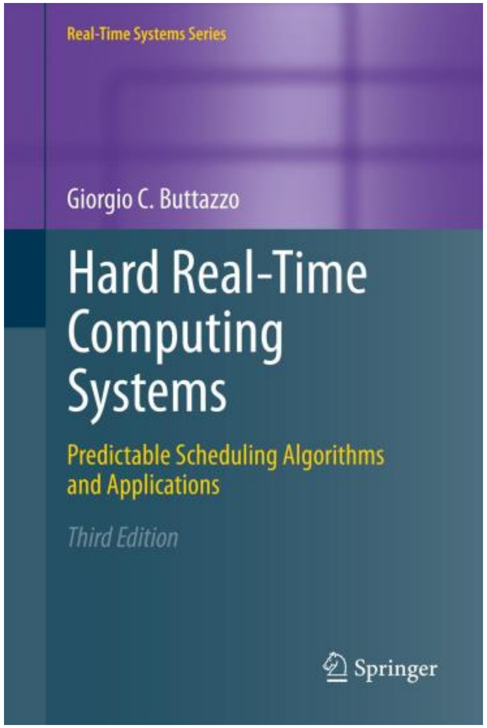
2IMN20 - Real-Time Systems

Schedulability analysis of EDF

Geoffrey Nelissen

Reference book

Chapter 4



Disclaimer:
Many slides were provided by Dr. Mitra Nasri and Prof. Giorgio Buttazzo

Assumptions

- For this lecture, we assume

  A1. All jobs of τ_i execute for no more than C_i

  A2. Tasks are periodic or sporadic with no release jitter

  A3. Tasks are fully preemptive

  A4. Context switch, preemption, and scheduling overheads are zero

  A5. Tasks are independent:
  - no precedence relations
  - no resource constraints
  - no blocking on I/O operations

  A6. No self-suspensions

  A7. Single core

EDF

Schedulability test

- **Liu and Layland 1973**

  A task set τ is schedulable by EDF **if and only if:**

$$\sum_{i=1}^{n} \frac{C_i}{T_i} \leq 1$$

Why is this not enough?

EDF

Schedulability test

- **Liu and Layland 1973**

  A task set τ is schedulable by EDF **if and only if:**

$$\sum_{i=1}^{n} \frac{C_i}{T_i} \leq 1$$

Assumption:
$\forall \boldsymbol{i}, \boldsymbol{D_i} = \boldsymbol{T_i}$

What if $D_i \neq T_i$?

EDF schedulability analysis for $D_i < T_i$

Processor Demand Criterion [Baruah '90]

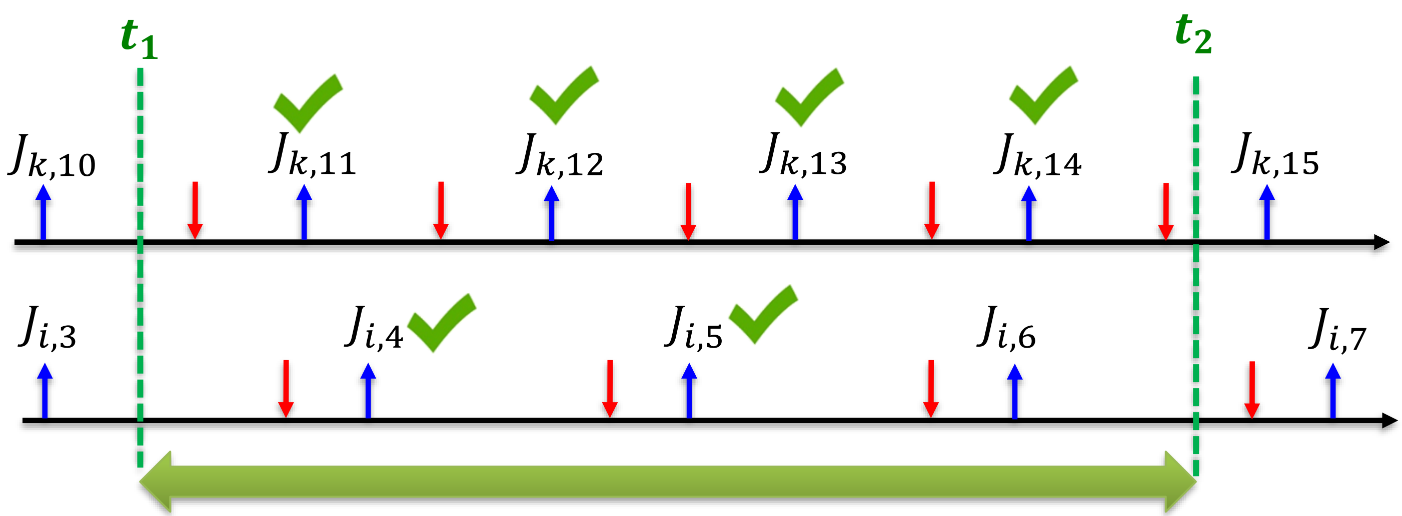
In **any interval** of length L, the **computational demand** $g(t, t+L)$ of the task set must be **no greater** than L.

$$\forall t, \forall L > 0, \qquad g(t, t+L) \leq L$$

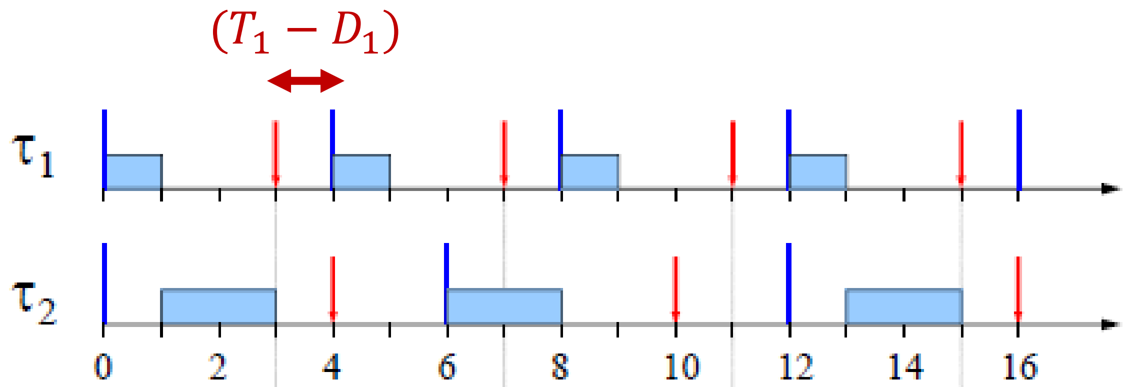
Understanding the processor demand (g function)

- **The demand** in $[t_1, t_2]$ is the computation time of those tasks arrived **at or after t_1** with deadline **less than or equal to t_2**:

$$g(t_1, t_2) = \sum_{a_{k,j} \geq t_1}^{d_{k,j} \leq t_2} C_k$$



Demand of a periodic/sporadic task set



Computational demand of τ_i

$$g_i(t, t+L) \leq g_i(0, L) = \left\lfloor \frac{L + (T_i - D_i)}{T_i} \right\rfloor \cdot C_i$$

$$= \left\lfloor \frac{L - D_i}{T_i} + 1 \right\rfloor \cdot C_i$$

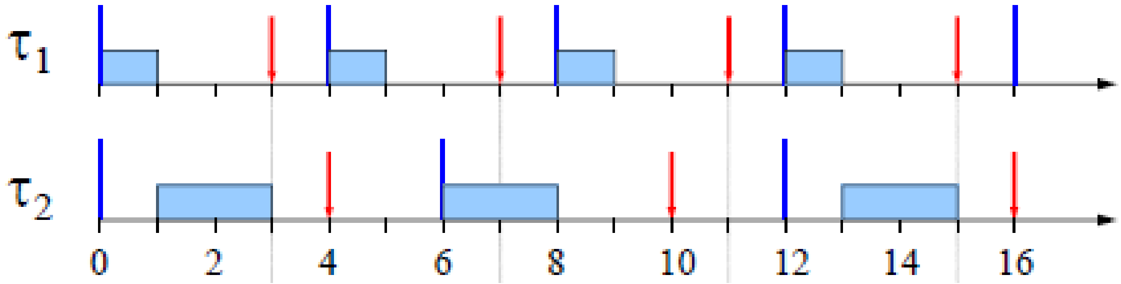
Total computational demand

$$\boxed{g(0, L) = \sum_{i=1}^{n} g_i(0, L)}$$

Applicable to sporadic tasks or periodic tasks without release jitter, i.e., $\forall i, \phi_i = 0$

Examples

$$g_i(0, L) = \left\lfloor \frac{L + (T_i - D_i)}{T_i} \right\rfloor \cdot C_i$$



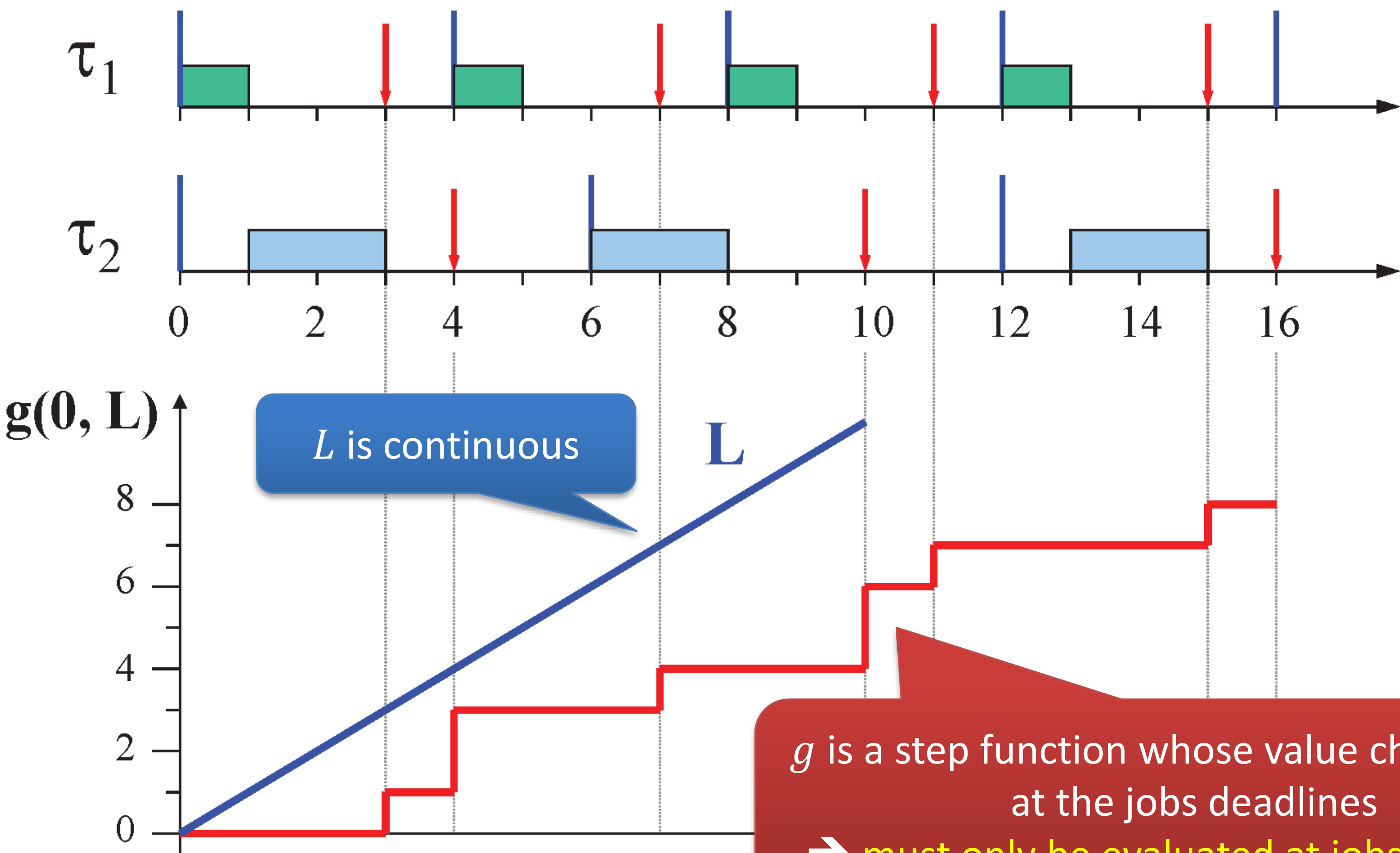
- What is $g_2(0, 4)$? $\quad g_2(0, 4) = \left\lfloor \frac{4 + 2}{6} \right\rfloor \cdot 2 = 2$
- What is $g_2(0, 8)$? $\quad g_2(0, 8) = \left\lfloor \frac{8 + 2}{6} \right\rfloor \cdot 2 = 2$
- What is $g_2(0, 16)$? $\quad g_2(0, 16) = \left\lfloor \frac{16 + 2}{6} \right\rfloor \cdot 2 = 6$

IRiS

Simplifying the test

$$\forall L, \qquad g_i(0, L) = \left\lfloor \frac{L + (T_i - D_i)}{T_i} \right\rfloor \cdot C_i \leq L$$

Testing all possible values of L is impractical



L is continuous

g is a step function whose value changes only at the jobs deadlines

➔ must only be evaluated at jobs deadlines

Processor Demand Test

$$\forall \boldsymbol{L} \in \mathcal{D}, \qquad \boldsymbol{g}(\boldsymbol{0}, \boldsymbol{L}) \leq \boldsymbol{L}$$

Where $\mathcal{D}$ is the set of deadline points for which $g(0, L)$ must be calculated:

$$\mathcal{D} = \{d_{i,j} \mid d_{i,j} \leq \min\{H, L^*\}\}$$

Deadlines are the only points where the value of $\boldsymbol{g}(\boldsymbol{0}, \boldsymbol{L})$ changes

$$H = lcm(T_1, \dots, T_n)$$

$$L^* = \frac{\sum_{i=1}^{n}(T_i - D_i) \cdot U_i}{1 - U}$$

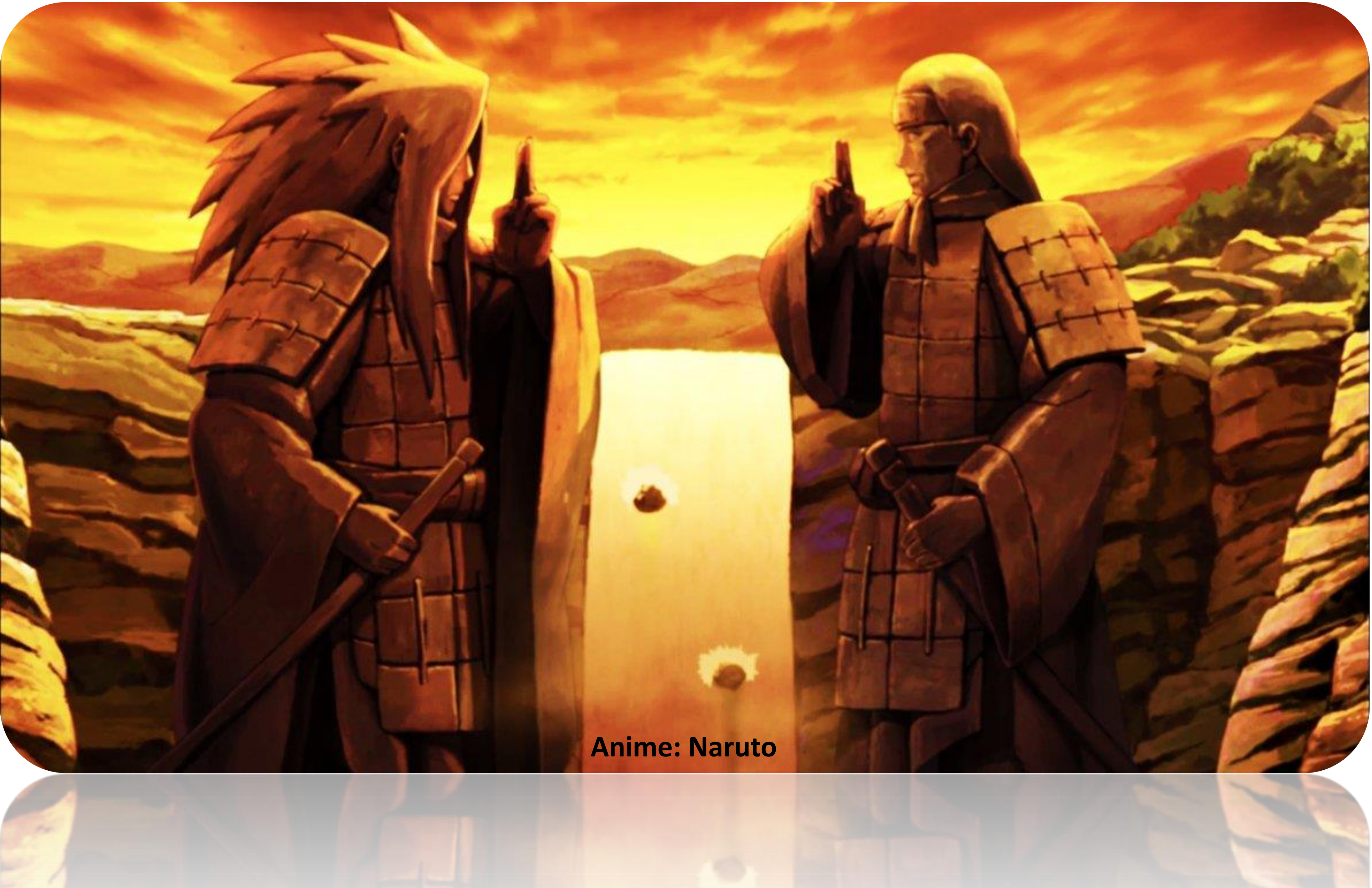
Two bounds on the maximum value of L that must be checked

Summary

Three analysis techniques:

- Processor utilization bound $\boldsymbol{U \leq U_{lb}}$
- Response time analysis $\forall i, \quad R_i \leq D_i$
- Processor demand criterion $\forall L > 0, \quad g(0, L) \leq L$

RM v.s. EDF

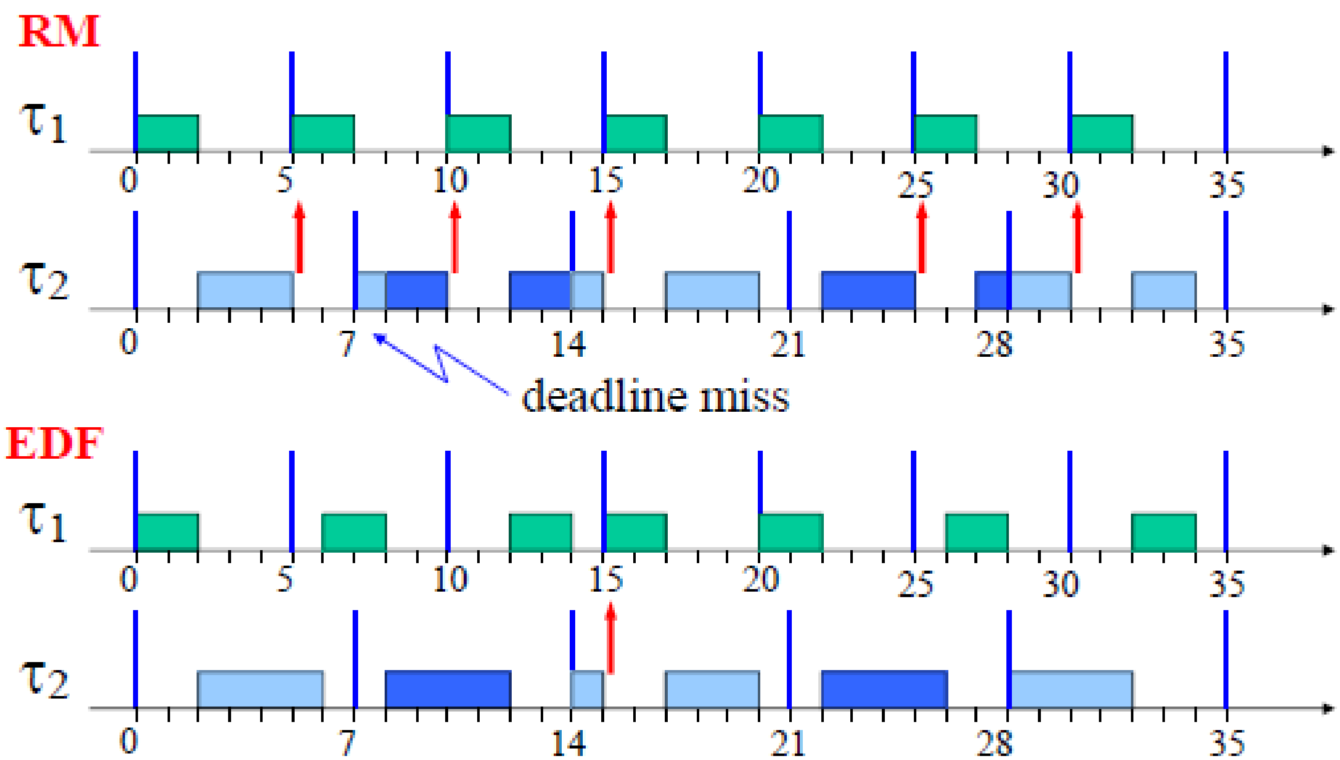


EDF vs FP

- **EDF is optimal** for independent tasks on single core, FP is not
- **FP is supported in almost all OS**, EDF is supported in a handful of them
- **FP is easier to implement** (lower runtime complexity)
- Runtime behavior may be simpler to explain with FP for non-experts

EDF vs RM

Context switches



In average, RM has more context switches

EDF vs RM

Schedulability analysis

	$D_i = T_i$	$D_i \leq T_i$
RM	*Suff.: polynomial* $O(n)$ LL: $\Sigma U_i \leq n(2^{1/n} - 1)$ HB: $\prod(U_i + 1) \leq 2$ *Exact* *pseudo-polynomial* RTA	*pseudo-polynomial* Response Time Analysis $\forall i \quad R_i \leq D_i$ $R_i = C_i + \sum_{k=1}^{i-1} \left\lceil \frac{R_i}{T_k} \right\rceil C_k$
EDF	*polynomial:* $O(n)$ $\Sigma U_i \leq 1$	*pseudo-polynomial* Processor Demand Analysis $\forall L > 0, \quad g(0, L) \leq L$

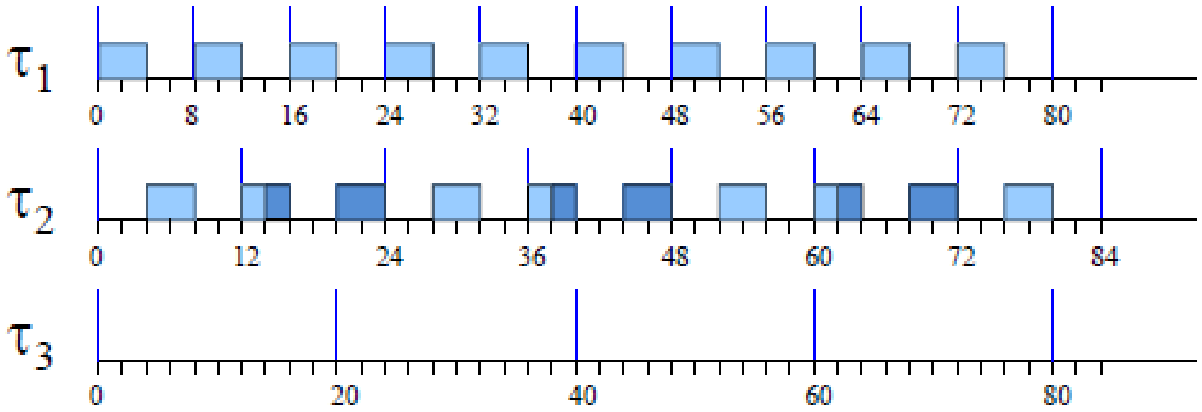
Not seen in the course but RM response-time analysis is faster than EDF

IRiS

EDF vs RM

RM under permanent overload: starvation for low priority tasks

$$U = \frac{4}{8} + \frac{6}{12} + \frac{5}{20} = 1.25$$

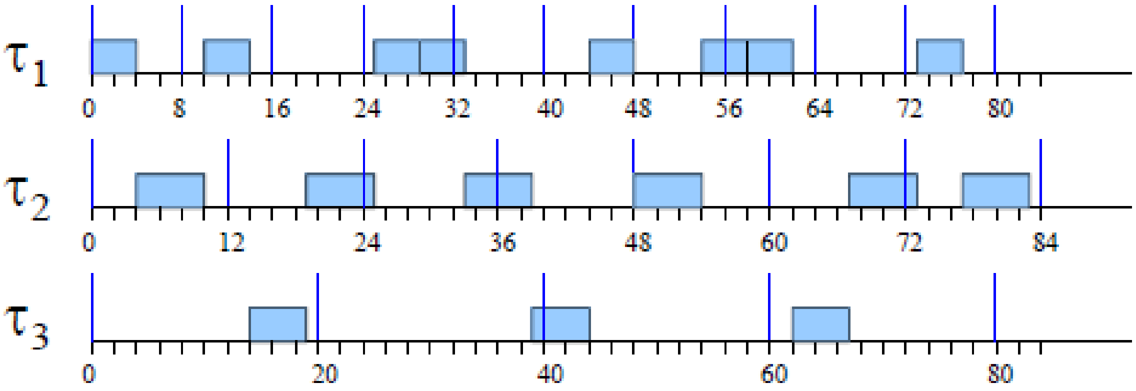


- High priority tasks execute at the proper rate
- Low priority tasks are completely blocked

EDF vs RM

EDF under permanent overload

$$U = \frac{4}{8} + \frac{6}{12} + \frac{5}{20} = 1.25$$



- All tasks execute at a slower rate
- No task is blocked