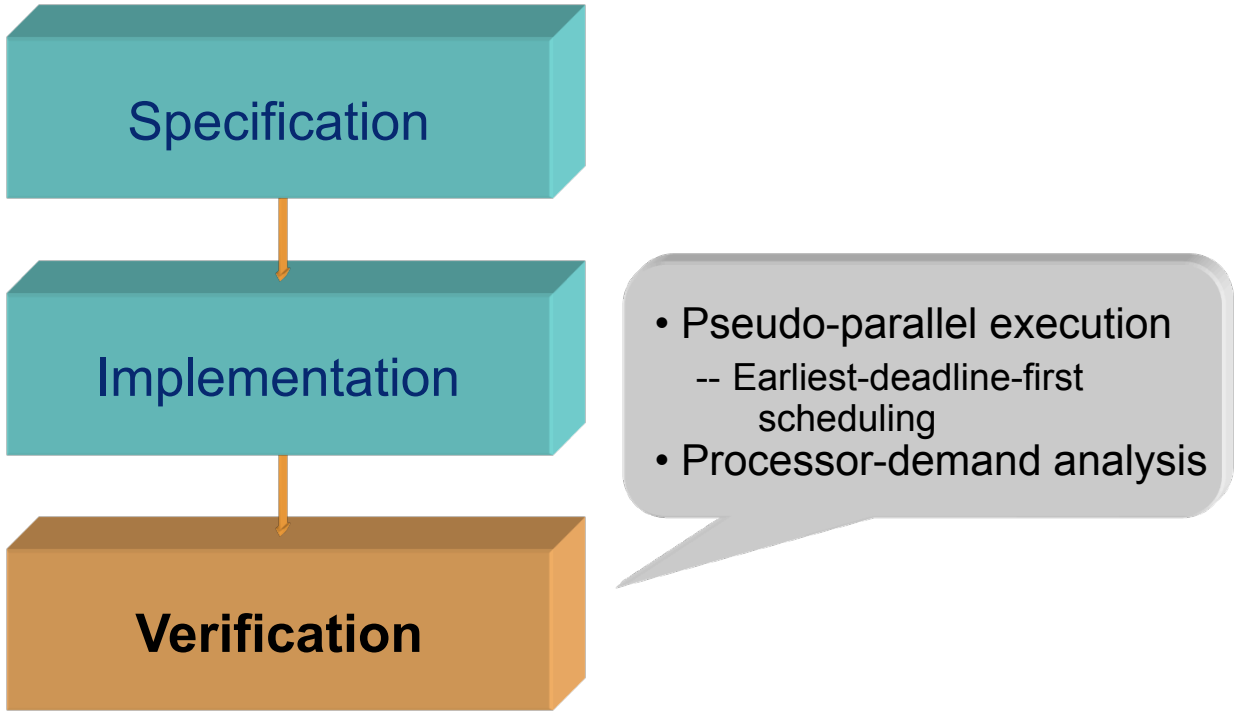
Real-Time Systems

Lecture #13

Professor Jan Jonsson

Department of Computer Science and Engineering
Chalmers University of Technology

Real-Time Systems



Example: scheduling using EDF

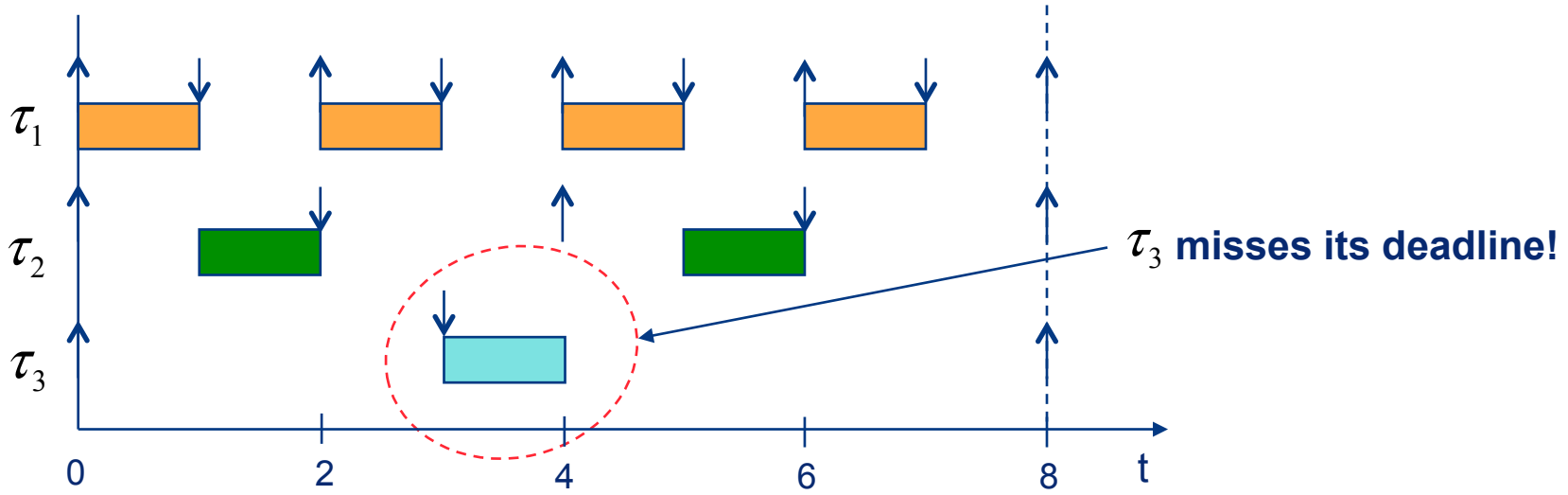
Problem: Assume a system with tasks according to the figure below. The timing properties of the tasks are given in the table. All tasks arrive the first time at time 0.

Investigate the schedulability of the tasks when EDF is used. (Note that $D_i < T_i$ for all tasks)

Task	C_i	D_i	T_i
τ_1	1	1	2
τ_2	1	2	4
τ_3	1	3	8

Example: scheduling using EDF

Simulate an execution of the tasks:



The tasks are not schedulable even though

$$U = \frac{1}{2} + \frac{1}{4} + \frac{1}{8} = \frac{7}{8} = 0.875 < 1$$

Feasibility analysis for EDF

What analysis methods are suitable for general EDF:

- Utilization-based analysis?
  Not suitable! Not general enough or exact enough
  - Does not work well for the case of $D_i < T_i$
- Response-time analysis?

  Not suitable! Analysis much more complex than for DM
  - Critical instant does not necessarily occur when all tasks arrive at the same time for the first time.
  - Instead, response time of a task is maximized at some scenario where all other tasks arrive at the same time; the worst such scenario has to be identified for each task before the response time of that task can be calculated.

Feasibility tests

What types of feasibility tests exist?

- Hyper period analysis (for any type of scheduler)
  - In an existing schedule no task execution may miss its deadline
- Processor utilization analysis (static/dynamic priority scheduling)
  - The fraction of processor time that is used for executing the task set must not exceed a given bound
- Response time analysis (static priority scheduling)
  - The worst-case response time for each task must not exceed the deadline of the task
- Processor demand analysis (dynamic priority scheduling)
  - The accumulated computation demand for the task set under a given time interval must not exceed the length of the interval

Processor-demand analysis

Processor demand:

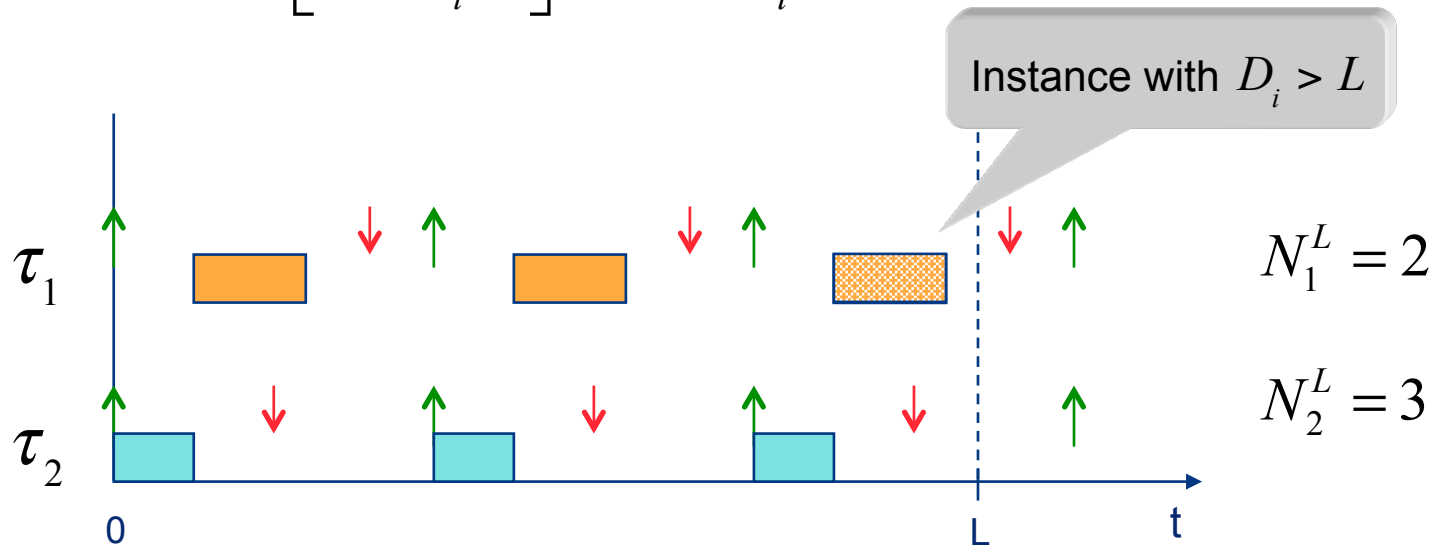
- The processor demand for a task τ_i in a given time interval $[0, L]$ is the amount of processor time that the task needs in the interval in order to meet the deadlines that fall within the interval.
- Let N_i^L represent the number of instances of τ_i that must complete execution before L .
- The total processor demand up to L is

$$C_P(0, L) = \sum_{i=1}^{n} N_i^L C_i$$

Processor-demand analysis

Processor demand:

- We can calculate N_i^L by counting how many times task τ_i has arrived during the interval $\left[0, L - D_i\right]$.
- We can ignore instances of the task that arrived during the interval $\left[L - D_i, L\right]$ since $D_i > L$ for these instances.



Processor-demand analysis

Processor-demand analysis:

- We can express N_i^L as

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

- The total processor demand is thus

$$C_P(0, L) = \sum_{i=1}^{n} \left(\left\lfloor \frac{L - D_i}{T_i} \right\rfloor + 1 \right) C_i$$

Exact feasibility test for EDF

(Sufficient and necessary condition)

A sufficient and necessary condition for EDF scheduling of synchronous task sets, for which $D_i \leq T_i$, is

$$\forall L : C_P(0, L) \leq L$$

where $C_P(0, L)$ is the total processor demand in $[0, L]$.

In other words: *for the task set to be schedulable with EDF there must not exist an interval of length L in the schedule where the processor demand in that interval exceeds the length L.*

The processor-demand analysis and associated feasibility test was presented by S. Baruah, L. Rosier and R. Howell in 1990.

Exact feasibility test for EDF

(Sufficient and necessary condition)

How many intervals must be examined?

- Only intervals coinciding with the absolute deadlines of tasks need to be examined
- The set of deadlines that need to be examined can be further reduced (see book excerpt)

What is the largest interval that must be examined?

- For synchronous task sets the largest interval can be bounded by the hyper period (LCM of task periods)

  ⇒ the analysis will in general have exponential time complexity
- For most synchronous task sets the largest interval to examine will be shorter than the hyper period. The analysis will then have pseudo-polynomial time complexity (see book excerpt)

Exact feasibility test for EDF

(Sufficient and necessary condition)

For synchronous task sets the feasibility test can consequently be rewritten as follows:

$$\forall L \in K : C_P(0, L) \leq L$$

$$K = \left\{ D_i^k \mid D_i^k = kT_i + D_i, D_i^k \leq \text{LCM}\left\{ T_1, \ldots, T_n \right\}, 1 \leq i \leq n, k \geq 0 \right\}$$

Exact feasibility test for EDF

(Sufficient and necessary condition)

The test is valid under the following assumptions:

1. All tasks are independent.
   - There must not exist dependencies due to precedence or mutual exclusion
2. All tasks are periodic or sporadic.
3. All tasks have identical offsets (= synchronous task set).
4. Task deadline does not exceed the period ($D_i \leq T_i$).
5. Task preemptions are allowed.

Feasibility tests

Summary

	$D_i = T_i$	$D_i \leq T_i$
Static priority (RM/DM)	$U \leq n(2^{1/n} - 1)$	$\forall i : R_i = C_i + \sum_{\forall j \in hp(i)} \left\lceil \frac{R_i}{T_j} \right\rceil C_j \leq D_i$
Dynamic priority (EDF)	$U \leq 1$	$\forall L : \sum_{i=1}^{n} \left(\left\lfloor \frac{L - D_i}{T_i} \right\rfloor + 1 \right) C_i \leq L$

Example 1: scheduling using EDF

Problem: We once again assume the system with tasks given in the beginning of this lecture.

Show, by using processor-demand analysis, that the tasks are not schedulable using EDF.

Task	C_i	D_i	T_i
τ_1	1	1	2
τ_2	1	2	4
τ_3	1	3	8

Example 1: scheduling using EDF

a) Determine the LCM for the tasks:

$$\mathrm{LCM}\{T_1, T_2, T_3\} = \mathrm{LCM}\{2,4,8\} = 8$$

Determine the control points K:

$$K_1 = \left\{ D_1^k \mid D_1^k = kT_1 + D_1,\ D_1^k \le 8,\ k = 0,1,2,3 \right\} = \{1,3,5,7\}$$

$$K_2 = \left\{ D_2^k \mid D_2^k = kT_2 + D_2,\ D_2^k \le 8,\ k = 0,1 \right\} = \{2,6\}$$

$$K_3 = \left\{ D_3^k \mid D_3^k = kT_3 + D_3,\ D_3^k \le 8,\ k = 0 \right\} = \{3\}$$

The processor demand must be checked at the following control points:

$$K = K_1 \cup K_2 \cup K_3 = \{1,2,3,5,6,7\}$$

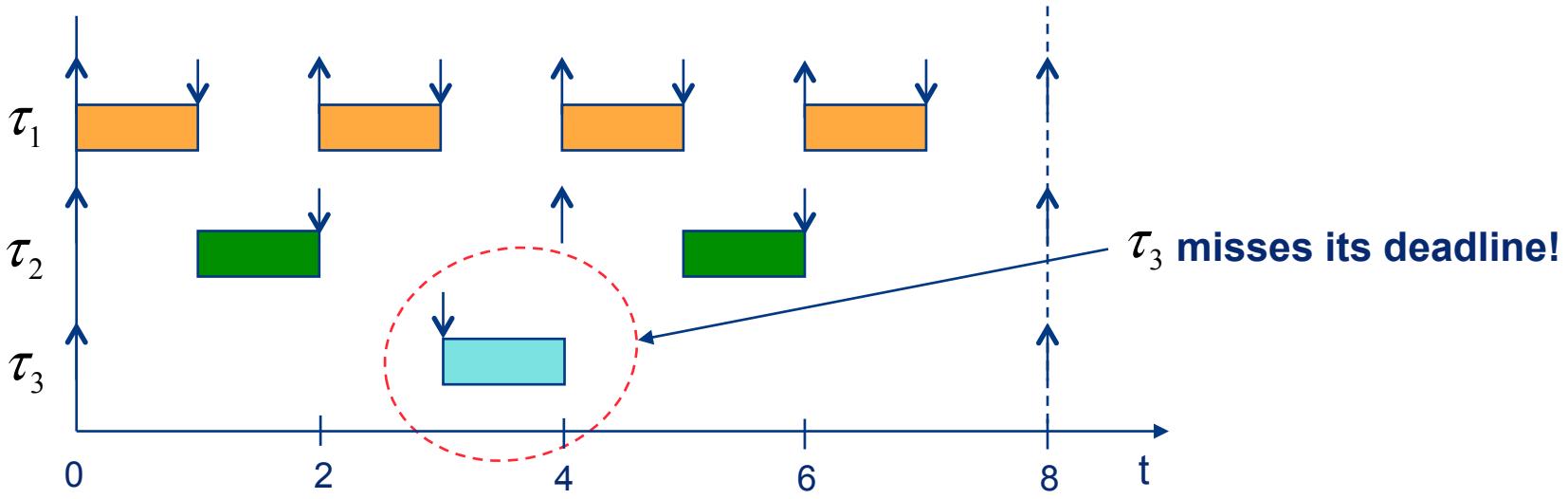
We define a table and examine every control point …

Example 1: scheduling using EDF

L	$N_1^L \cdot C_1$	$N_2^L \cdot C_2$	$N_3^L \cdot C_3$	$C_P(0,L)$	$C_P(0,L) \le L$
1	$\left(\left\lfloor \frac{1-1}{2} \right\rfloor + 1\right) \cdot 1 = 1$	$\left(\left\lfloor \frac{1-2}{4} \right\rfloor + 1\right) \cdot 1 = 0$	$\left(\left\lfloor \frac{1-3}{8} \right\rfloor + 1\right) \cdot 1 = 0$	1	OK!
2	$\left(\left\lfloor \frac{2-1}{2} \right\rfloor + 1\right) \cdot 1 = 1$	$\left(\left\lfloor \frac{2-2}{4} \right\rfloor + 1\right) \cdot 1 = 1$	$\left(\left\lfloor \frac{2-3}{8} \right\rfloor + 1\right) \cdot 1 = 0$	2	OK!
3	$\left(\left\lfloor \frac{3-1}{2} \right\rfloor + 1\right) \cdot 1 = 2$	$\left(\left\lfloor \frac{3-2}{4} \right\rfloor + 1\right) \cdot 1 = 1$	$\left(\left\lfloor \frac{3-3}{8} \right\rfloor + 1\right) \cdot 1 = 1$	4	Not OK!
5	$\left(\left\lfloor \frac{5-1}{2} \right\rfloor + 1\right) \cdot 1 = 3$	$\left(\left\lfloor \frac{5-2}{4} \right\rfloor + 1\right) \cdot 1 = 1$	$\left(\left\lfloor \frac{5-3}{8} \right\rfloor + 1\right) \cdot 1 = 1$	5	OK!
6	$\left(\left\lfloor \frac{6-1}{2} \right\rfloor + 1\right) \cdot 1 = 3$	$\left(\left\lfloor \frac{6-2}{4} \right\rfloor + 1\right) \cdot 1 = 2$	$\left(\left\lfloor \frac{6-3}{8} \right\rfloor + 1\right) \cdot 1 = 1$	6	OK!
7	$\left(\left\lfloor \frac{7-1}{2} \right\rfloor + 1\right) \cdot 1 = 4$	$\left(\left\lfloor \frac{7-2}{4} \right\rfloor + 1\right) \cdot 1 = 2$	$\left(\left\lfloor \frac{7-3}{8} \right\rfloor + 1\right) \cdot 1 = 1$	7	OK!

Example 1: scheduling using EDF

As we saw in the beginning of the lecture the resulting schedule looks like this:



Extended processor-demand analysis

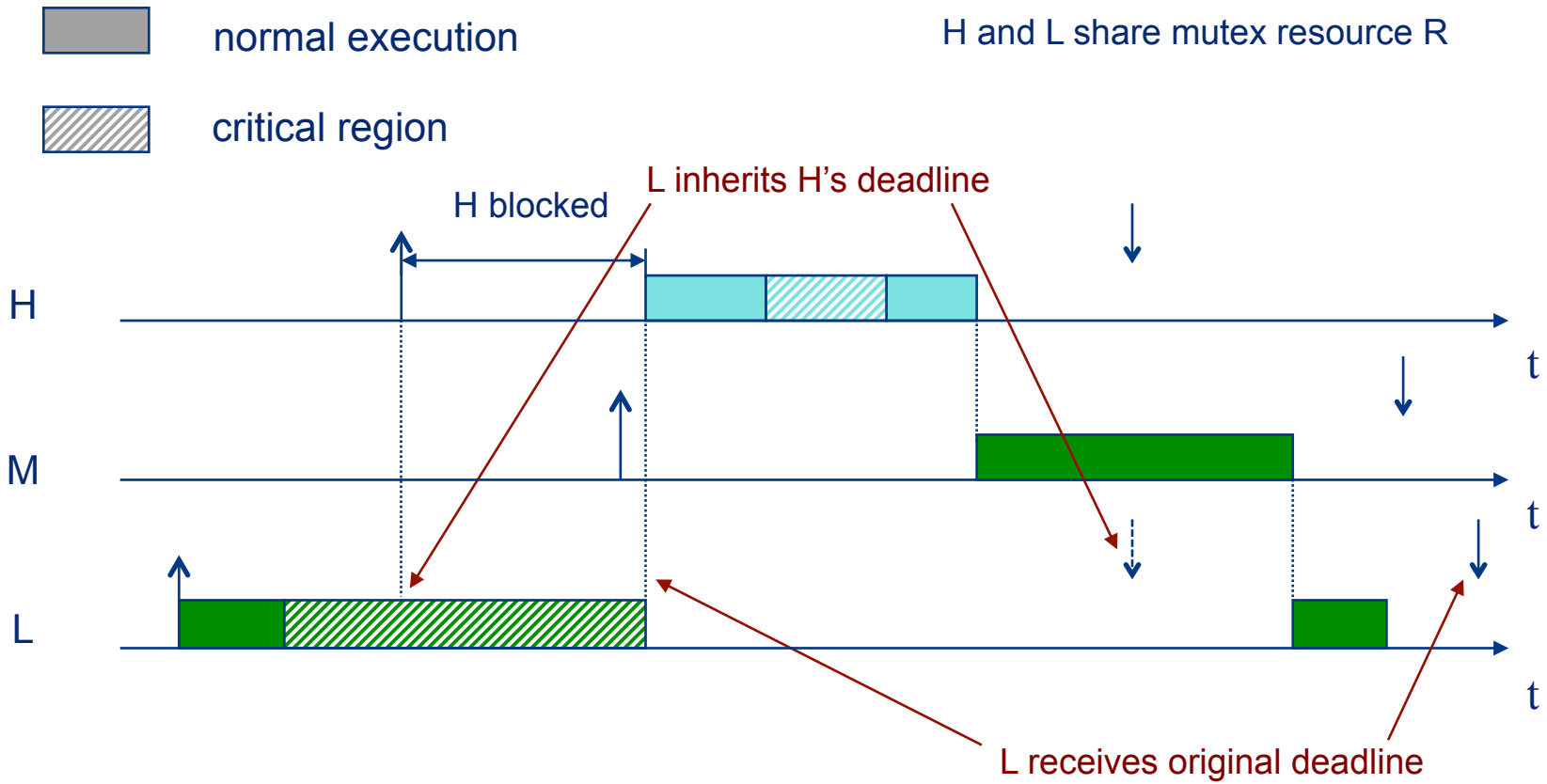
The feasibility test can be extended to handle:

- Blocking
- Start-time variations (”release jitter”)
- Time offsets (asynchronous task sets)
- …

In this course, we only briefly describe how blocking is handled.

Recollection from an earlier lecture

Blocking using deadline inheritance protocol (DIP):



Extended processor-demand analysis

Accounting for blocking with EDF:

- Which tasks can block the task being analyzed?

  Dynamic relationship! Depends on which tasks with longer deadlines are active when the task is ready for execution.

  – With static priorities (RM, DM) this was much easier to determine

- Simple method is the Deadline Inheritance Protocol (DIP):

  – When a task blocks one or more tasks with deadlines closer in time, it temporarily assumes (inherits) the deadline closest in time of the blocked tasks.

  – Due to the dynamic relationship, and due to the possibility of chained blocking, the estimated worst-case blocking times will be very pessimistic.

Extended processor-demand analysis

Accounting for blocking with EDF:

- The best method is the Stack Resource Policy (SRP):

  Very similar to ICPP used for static priority scheduling.
  - Each task is assigned a static preemption level, that reflects the relative deadline of the task.
  - Each shared resource is given a ceiling value based on the maximum preemption level of the tasks that use the resource.
  - A task can only preempt the currently-executing task if its deadline is closer in time, and its preemption level is higher than the highest ceiling of the currently locked resources.
  - Due to the protocol, and because no chained blocking occurs, accurate worst-case blocking times can determined and be incorporated into the processor-demand analysis.

Example 2: scheduling using EDF

Problem: Assume a system with tasks according to the figure below. The timing properties of the tasks are given in the table.

a) Determine, by using processor-demand analysis, whether the tasks are schedulable or not using EDF.

b) Determine, by using simulation, whether the tasks are schedulable or not using EDF.

Task	C_i	D_i	T_i
τ_1	2	3	4
τ_2	2	7	8
τ_3	3	12	16

Example 2: scheduling using EDF

a) Determine the LCM for the tasks:

$$\mathrm{LCM}\{T_1, T_2, T_3\} = \mathrm{LCM}\{4, 8, 16\} = 16$$

Determine the control points K:

$$K_1 = \left\{ D_1^k \mid D_1^k = kT_1 + D_1,\ D_1^k \le 16,\ k = 0,1,2,3 \right\} = \{3, 7, 11, 15\}$$

$$K_2 = \left\{ D_2^k \mid D_2^k = kT_2 + D_2,\ D_2^k \le 16,\ k = 0,1 \right\} = \{7, 15\}$$

$$K_3 = \left\{ D_3^k \mid D_3^k = kT_3 + D_3,\ D_3^k \le 16,\ k = 0 \right\} = \{12\}$$

The processor demand must be checked at the following control points:

$$K = K_1 \cup K_2 \cup K_3 = \{3, 7, 11, 12, 15\}$$

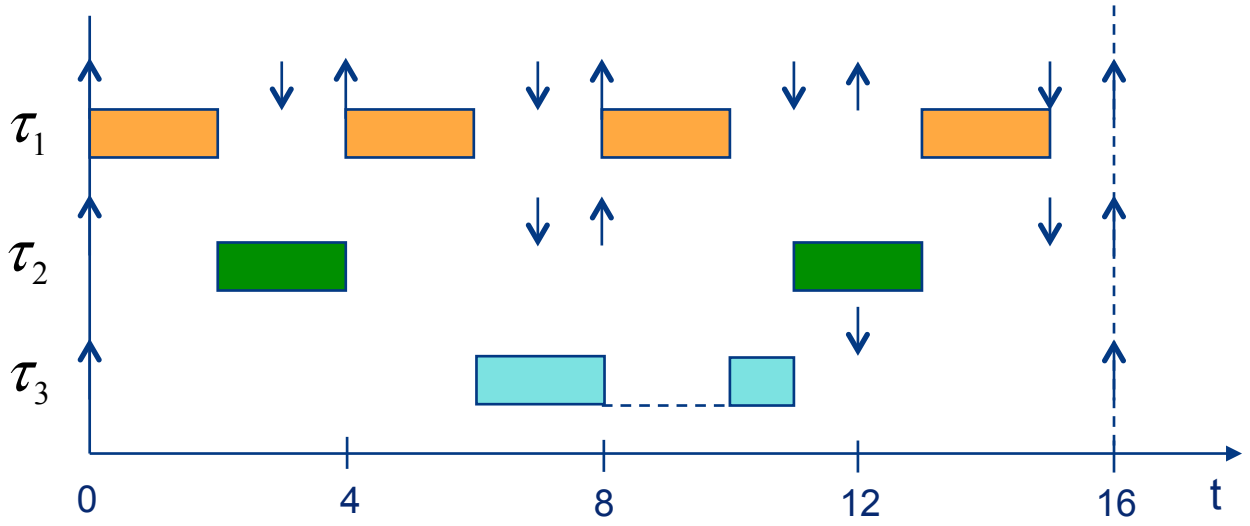
We define a table and examine every control point …

Example 2: scheduling using EDF

L	$N_1^L \cdot C_1$	$N_2^L \cdot C_2$	$N_3^L \cdot C_3$	$C_P(0,L)$	$C_P(0,L) \le L$
3	$\left(\left\lfloor \frac{3-3}{4} \right\rfloor + 1\right) \cdot 2 = 2$	$\left(\left\lfloor \frac{3-7}{8} \right\rfloor + 1\right) \cdot 2 = 0$	$\left(\left\lfloor \frac{3-12}{16} \right\rfloor + 1\right) \cdot 3 = 0$	2	OK!
7	$\left(\left\lfloor \frac{7-3}{4} \right\rfloor + 1\right) \cdot 2 = 4$	$\left(\left\lfloor \frac{7-7}{8} \right\rfloor + 1\right) \cdot 2 = 2$	$\left(\left\lfloor \frac{7-12}{16} \right\rfloor + 1\right) \cdot 3 = 0$	6	OK!
11	$\left(\left\lfloor \frac{11-3}{4} \right\rfloor + 1\right) \cdot 2 = 6$	$\left(\left\lfloor \frac{11-7}{8} \right\rfloor + 1\right) \cdot 2 = 2$	$\left(\left\lfloor \frac{11-12}{16} \right\rfloor + 1\right) \cdot 3 = 0$	8	OK!
12	$\left(\left\lfloor \frac{12-3}{4} \right\rfloor + 1\right) \cdot 2 = 6$	$\left(\left\lfloor \frac{12-7}{8} \right\rfloor + 1\right) \cdot 2 = 2$	$\left(\left\lfloor \frac{12-12}{16} \right\rfloor + 1\right) \cdot 3 = 3$	11	OK!
15	$\left(\left\lfloor \frac{15-3}{4} \right\rfloor + 1\right) \cdot 2 = 8$	$\left(\left\lfloor \frac{15-7}{8} \right\rfloor + 1\right) \cdot 2 = 4$	$\left(\left\lfloor \frac{15-12}{16} \right\rfloor + 1\right) \cdot 3 = 3$	15	OK!

Example 2: scheduling using EDF

b) Simulate the execution of the tasks:



The tasks meet their deadlines also in this case!