



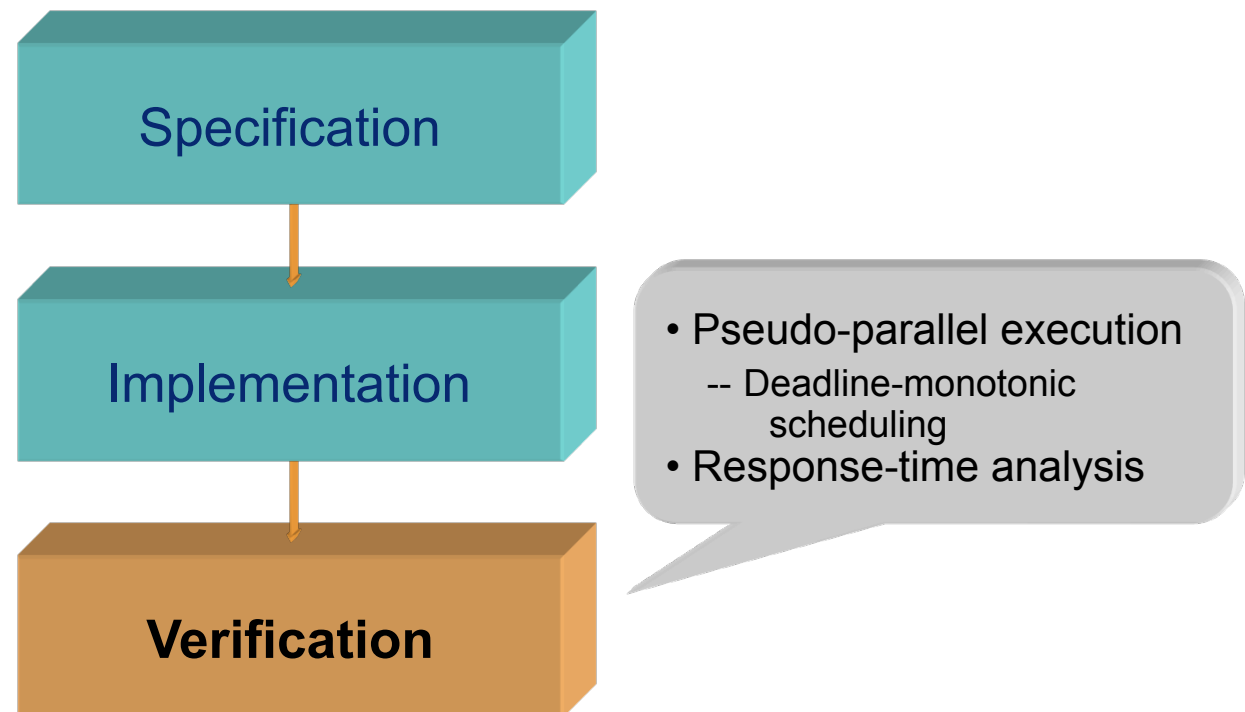
# Real-Time Systems

## Lecture #12

Professor Jan Jonsson

Department of Computer Science and Engineering  
Chalmers University of Technology

# Real-Time Systems



# Example: scheduling using RM

**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.

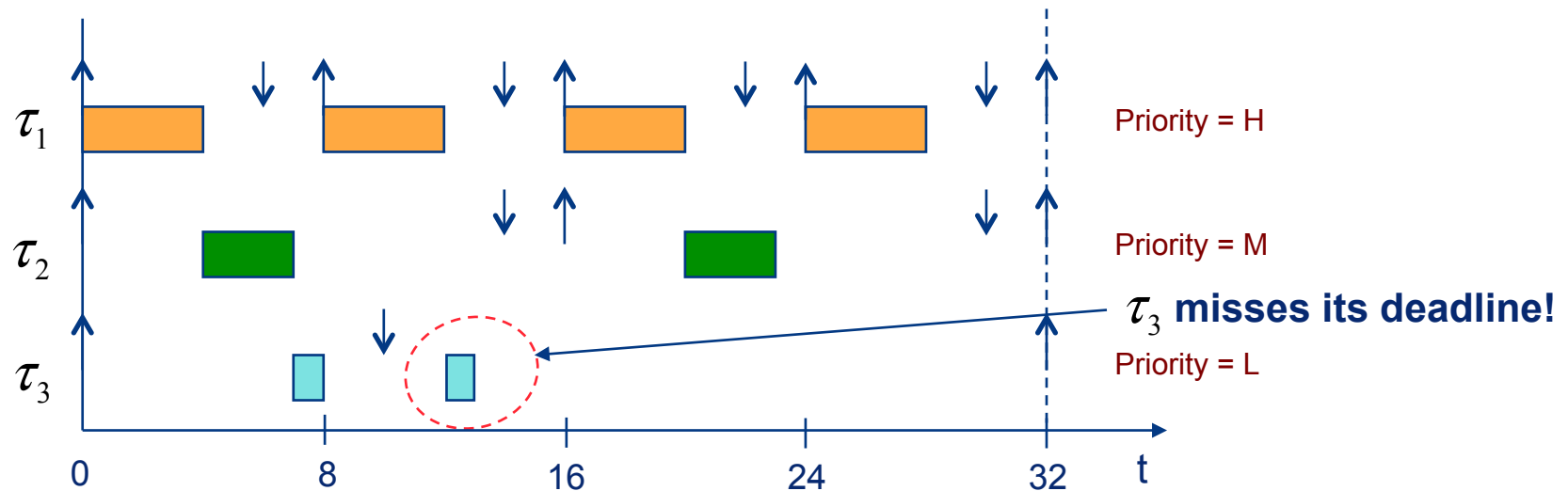
Investigate the schedulability of the tasks when RM is used.  
(Note that  $D_i < T_i$  for all tasks)



| Task     | $C_i$ | $D_i$ | $T_i$ |
|----------|-------|-------|-------|
| $\tau_1$ | 4     | 6     | 8     |
| $\tau_2$ | 3     | 14    | 16    |
| $\tau_3$ | 2     | 10    | 32    |

# Example: scheduling using RM

Simulate an execution of the tasks using RM:



The tasks are not schedulable even though

$$U = \frac{4}{8} + \frac{3}{16} + \frac{2}{32} = \frac{24}{32} = 0.75 < U_{RM} = 3(2^{1/3} - 1) \approx 0.780$$

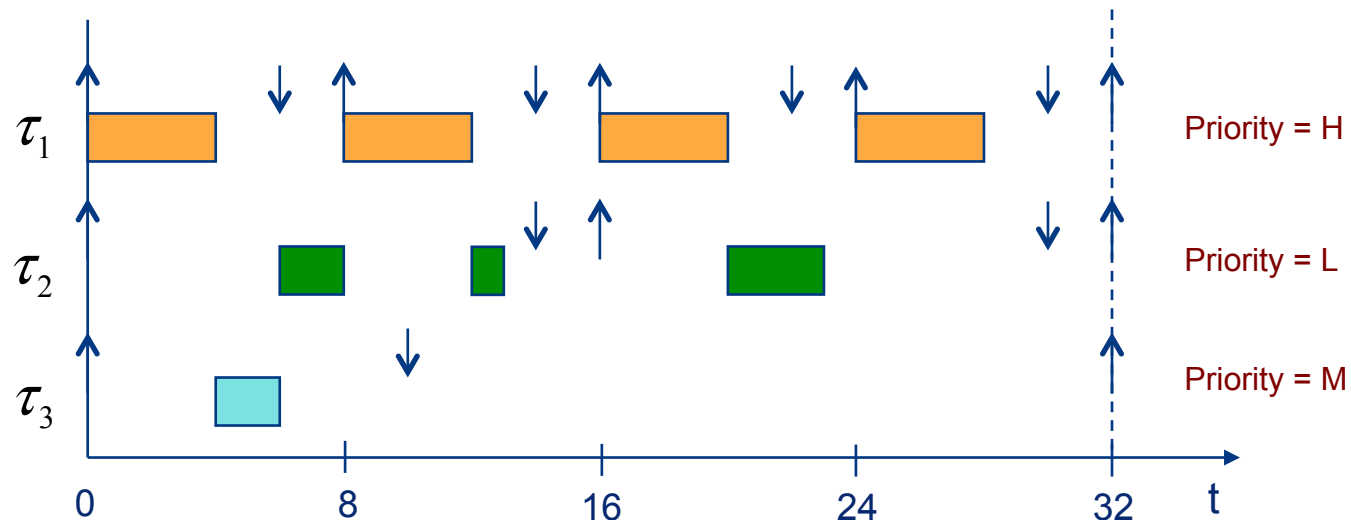
# Deadline-monotonic scheduling

## Properties:

- Uses static priorities
  - Priority is determined by urgency: the task with the shortest relative deadline receives highest priority
  - Proposed as a generalization of rate-monotonic scheduling (RM is a special case of DM, with  $D_i = T_i$ )
- Theoretically well-established
  - Exact feasibility test is an NP-complete problem (pseudo-polynomial time with response-time analysis)
  - DM is optimal among all scheduling algorithms that use static task priorities for which  $D_i \leq T_i$  (shown by J. Leung and J. W. Whitehead in 1982)

# Example: scheduling using DM

Simulate an execution of the task set given earlier using DM:



All tasks now meet their deadlines!

# 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

# Response-time analysis

The response time  $R_i$  for a task  $\tau_i$  represents the worst-case completion time of the task when execution interference from other tasks are accounted for.

The response time for a task  $\tau_i$  consists of:

$C_i$  The task's uninterrupted execution time (WCET)

$I_i$  Interference from higher-priority tasks

$$R_i = C_i + I_i$$

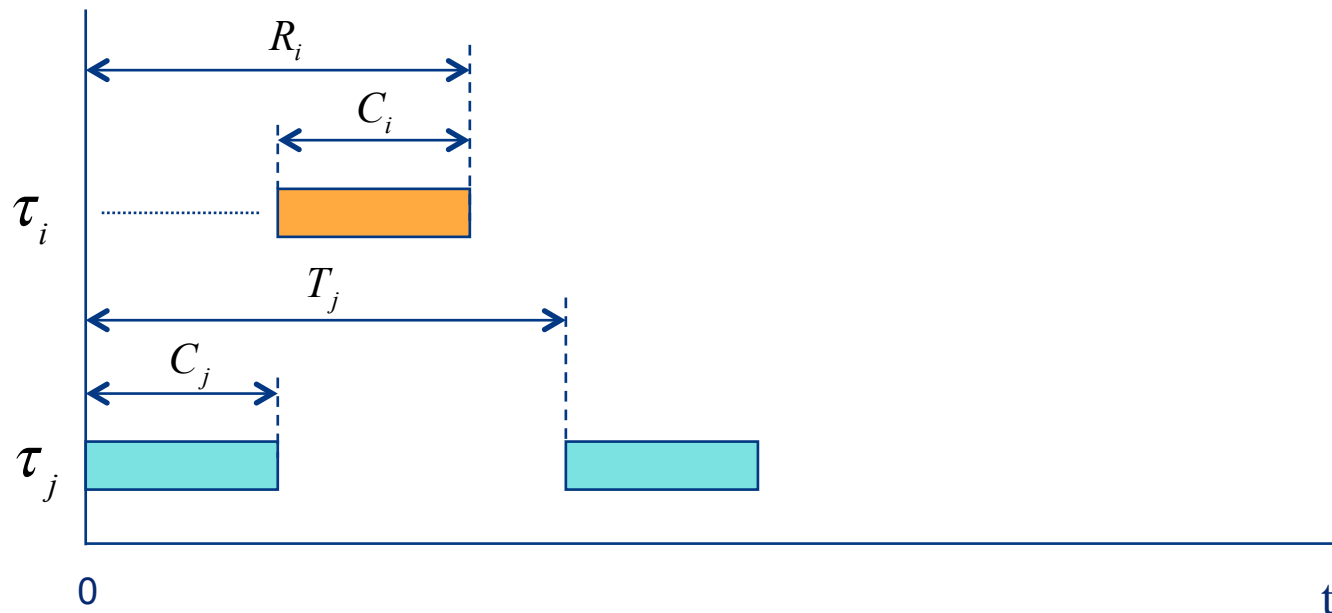


# Response-time analysis

## Interference:

Consider two tasks,  $\tau_i$  and  $\tau_j$ , where  $\tau_j$  has higher priority

Case 1:  $0 < R_i \leq T_j \Rightarrow R_i = C_i + C_j$

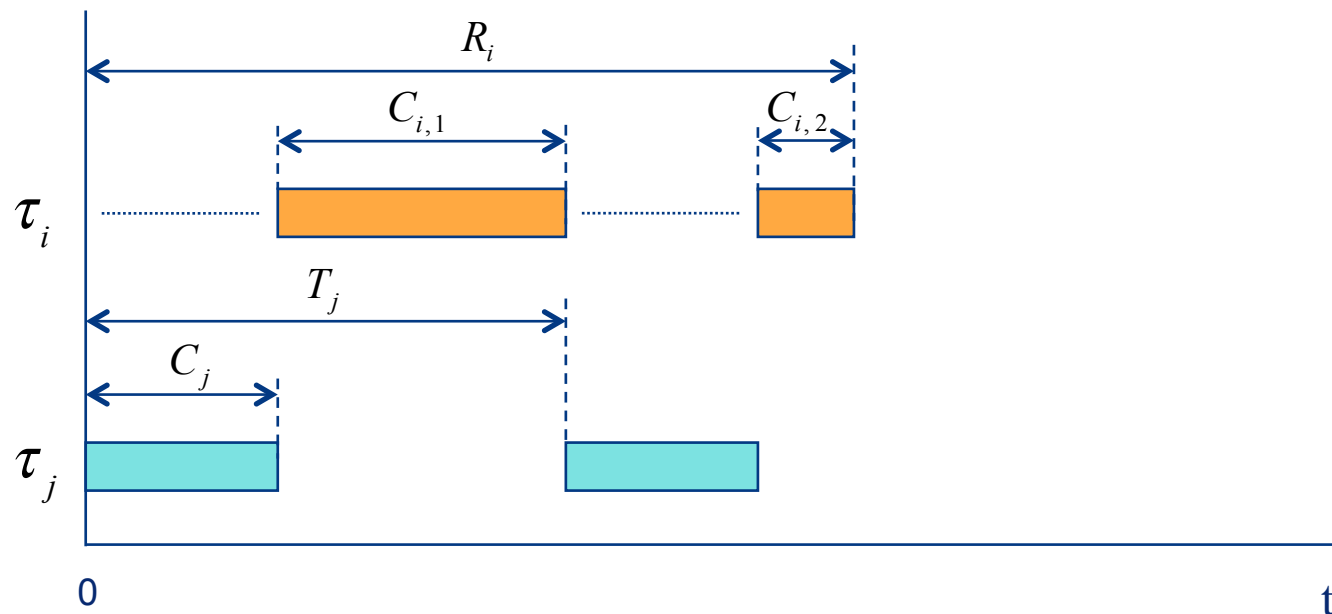


# Response-time analysis

## Interference:

Consider two tasks,  $\tau_i$  and  $\tau_j$ , where  $\tau_j$  has higher priority

Case 2:  $T_j < R_i \leq 2T_j \Rightarrow R_i = C_i + 2C_j$



# Response-time analysis

## Interference:

When task  $\tau_i$  is preempted by higher-priority task  $\tau_j$ :

The response time for  $\tau_i$  is at most  $R_i$  time units.

If  $0 < R_i \leq T_j$ , task  $\tau_i$  can be preempted at most one time by  $\tau_j$

If  $T_j < R_i \leq 2T_j$ , task  $\tau_i$  can be preempted at most two times by  $\tau_j$

If  $2T_j < R_i \leq 3T_j$ , task  $\tau_i$  can be preempted at most three times by  $\tau_j$

...

The number of interferences from  $\tau_j$  is thus limited by:  $\left\lceil \frac{R_i}{T_j} \right\rceil$

The total time for these interferences are:  $\left\lceil \frac{R_i}{T_j} \right\rceil C_j$

# Response-time analysis

## Interference:

- For static-priority scheduling, the interference term is

$$I_i = \sum_{\forall j \in hp(i)} \left\lceil \frac{R_i}{T_j} \right\rceil C_j$$

where  $hp(i)$  is the set of tasks with higher priority than  $\tau_i$ .

- The response time for a task  $\tau_i$  is thus:

$$R_i = C_i + \sum_{\forall j \in hp(i)} \left\lceil \frac{R_i}{T_j} \right\rceil C_j$$

# Response-time analysis

## Interference:

- The equation does not have a simple analytic solution.
- However, an iterative procedure can be used:

$$R_i^{n+1} = C_i + \sum_{\forall j \in hp(i)} \left\lceil \frac{R_i^n}{T_j} \right\rceil C_j$$

- The iteration starts with a value that is guaranteed to be less than or equal to the final value of  $R_i$  (e.g.  $R_i^0 = C_i$ )
- The iteration completes at convergence ( $R_i^{n+1} = R_i^n$ ) or if the response time exceeds some threshold (e.g.  $D_i$ )

# Exact feasibility test for DM

(Sufficient and necessary condition)

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

$$\forall i: R_i \leq D_i$$

where  $R_i$  is the worst-case response time for task  $\tau_i$

*In other words: for the task set to be schedulable with DM there must not exist an instance of a task execution in the schedule where the worst-case response time of the task exceeds its deadline.*

The response-time analysis and associated feasibility test was presented by M. Joseph and P. Pandya in 1986.

# Exact feasibility test for DM

(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.

# Example 1: scheduling using DM

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

Show, by using response-time analysis, that the tasks are schedulable using DM.



| Task     | $C_i$ | $D_i$ | $T_i$ |
|----------|-------|-------|-------|
| $\tau_1$ | 4     | 6     | 8     |
| $\tau_2$ | 3     | 14    | 16    |
| $\tau_3$ | 2     | 10    | 32    |



# Example 1: scheduling using DM

Calculation of response times:

[  $\tau_1$  has highest priority w r t DM ]

$$R_1 = C_1 = 4 \leq D_1 = 6 \Rightarrow \text{OK!}$$

[  $\tau_3$  has medium priority w r t DM ]

$$R_3 = C_3 + \left\lceil \frac{R_3}{T_1} \right\rceil C_1 \quad [\text{Assume } R_3^0 = C_3 = 2]$$

$$R_3^1 = 2 + \left\lceil \frac{2}{8} \right\rceil \cdot 4 = 2 + 1 \cdot 4 = 6$$

$$R_3^2 = 2 + \left\lceil \frac{6}{8} \right\rceil \cdot 4 = 2 + 1 \cdot 4 = 6 \quad [\text{Convergence because } R_3^2 = R_3^1]$$

$$\leq D_3 = 10 \Rightarrow \text{OK!}$$

# Example 1: scheduling using DM

[  $\tau_2$  has lowest priority w r t DM ]

$$R_2 = C_2 + \left\lceil \frac{R_2}{T_1} \right\rceil C_1 + \left\lceil \frac{R_2}{T_3} \right\rceil C_3 \quad [ \text{Assume } R_2^0 = C_2 = 3 ]$$

$$R_2^1 = 3 + \left\lceil \frac{3}{8} \right\rceil \cdot 4 + \left\lceil \frac{3}{32} \right\rceil \cdot 2 = 3 + 1 \cdot 4 + 1 \cdot 2 = 9$$

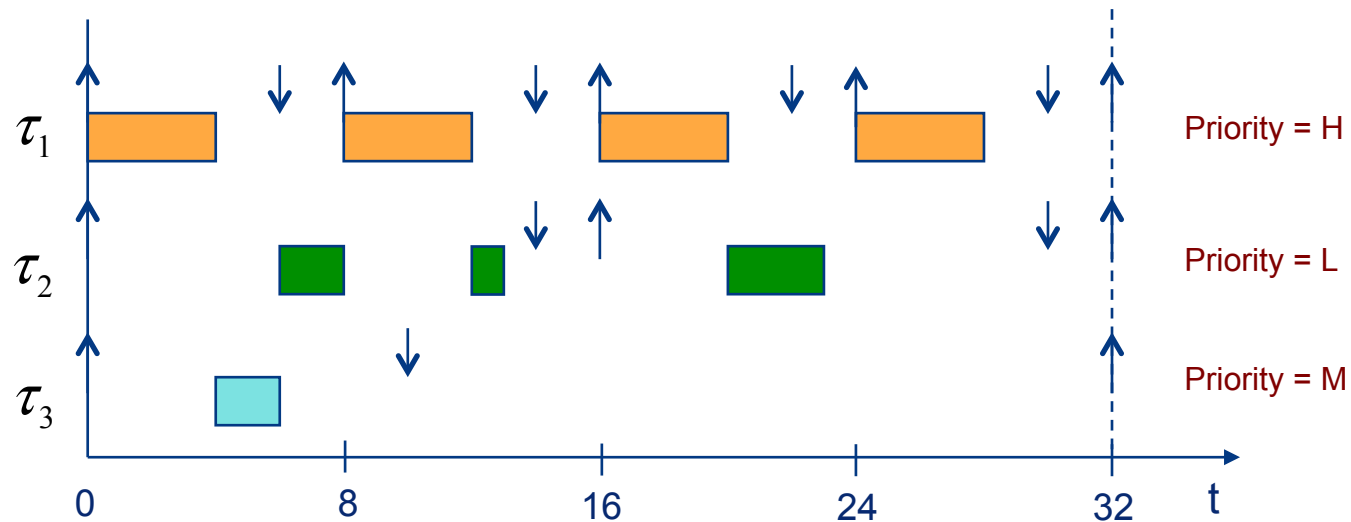
$$R_2^2 = 3 + \left\lceil \frac{9}{8} \right\rceil \cdot 4 + \left\lceil \frac{9}{32} \right\rceil \cdot 2 = 3 + 2 \cdot 4 + 1 \cdot 2 = 13$$

[ Convergence because  $R_2^3 = R_2^2$  ]

$$R_2^3 = 3 + \left\lceil \frac{13}{8} \right\rceil \cdot 4 + \left\lceil \frac{13}{32} \right\rceil \cdot 2 = 3 + 2 \cdot 4 + 1 \cdot 2 = 13 \leq D_2 = 14 \Rightarrow \text{OK!}$$

# Example 1: scheduling using DM

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



# Extended response-time analysis

The test can be extended to handle:

- Blocking
- Start-time variations ("release jitter")
- Time offsets (asynchronous task sets)
- Deadlines exceeding the period
- Overhead due to context switches, timers, interrupts, ...

In this course, we only show how blocking is handled.

# Extended response-time analysis

Blocking can be accounted for in the following cases:

- Blocking caused by critical regions
  - Blocking factor  $B_i$  represents the length of critical region(s) that are executed by tasks with lower priority than  $\tau_i$
- Blocking caused by non-preemptive scheduling
  - Blocking factor  $B_i$  represents largest WCET (not counting  $\tau_i$ )

$$R_i = C_i + B_i + \sum_{\forall j \in hp(i)} \left\lceil \frac{R_i}{T_j} \right\rceil C_j$$

# Recollection from an earlier lecture

## Priority Ceiling Protocol:

- Basic idea:

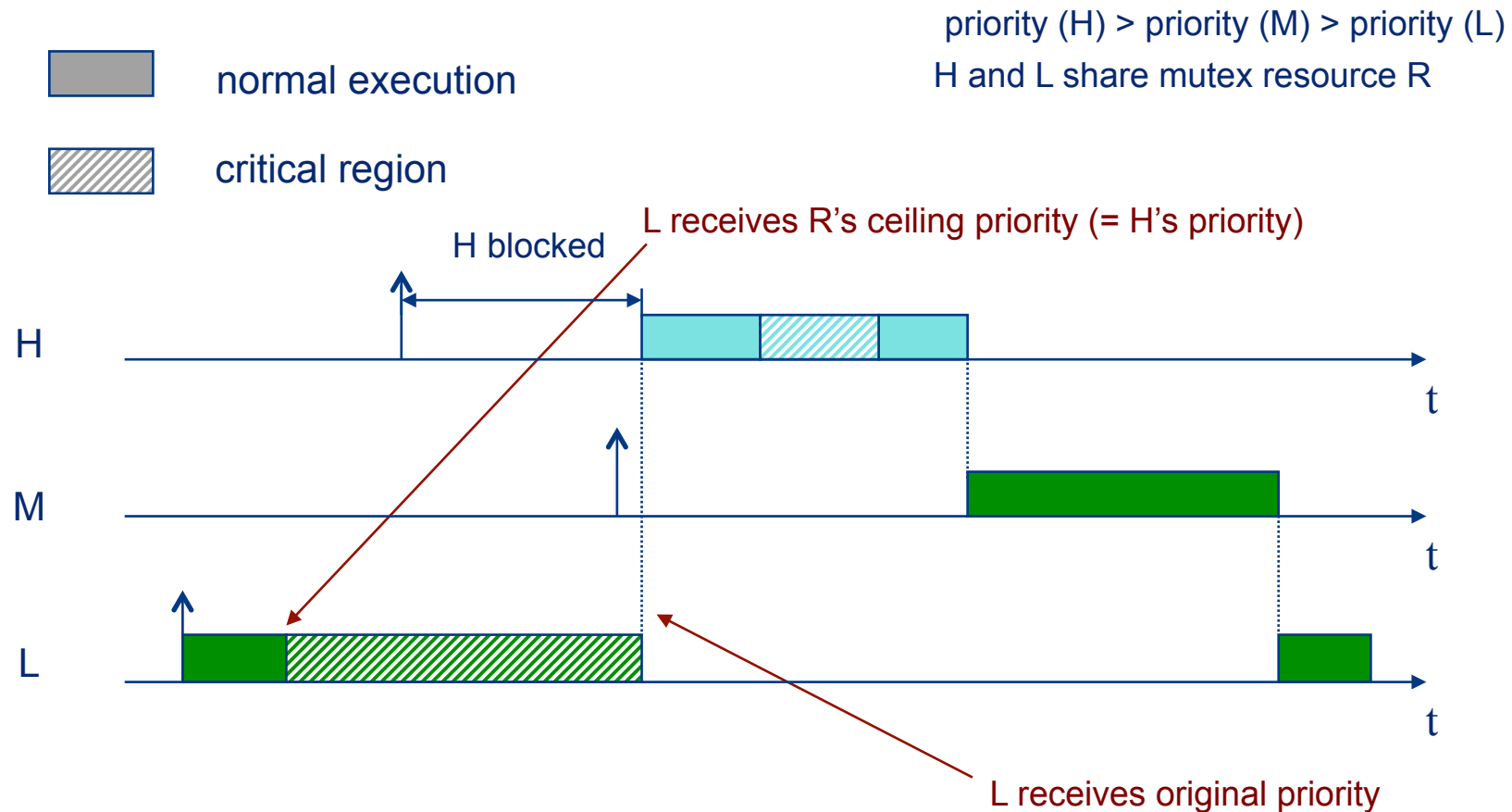
Each resource is assigned a priority ceiling equal to the priority of the highest-priority task that can lock it.

Then, a task  $\tau_i$  is allowed to enter a critical region only if its priority is higher than all priority ceilings of the resources currently locked by tasks other than  $\tau_i$ .

When a task  $\tau_i$  blocks one or more higher-priority tasks, it temporarily inherits the highest priority of the blocked tasks.

# Recollection from an earlier lecture

## Blocking using ceiling priority protocol ICPP:



# Extended response-time analysis

## Blocking caused by lower-priority tasks:

- When using a priority ceiling protocol (such as ICPP), a task  $\tau_i$  can only be blocked once by a task with lower priority than  $\tau_i$ .
- This occurs if the lower-priority task is within a critical region when  $\tau_i$  arrives, and the critical region's ceiling priority is higher than or equal to the priority of  $\tau_i$ .
- Blocking now means that the start time of  $\tau_i$  is delayed (= the blocking factor  $B_i$ )
- As soon as  $\tau_i$  has started its execution, it cannot be blocked by a lower-priority task.



# Extended response-time analysis

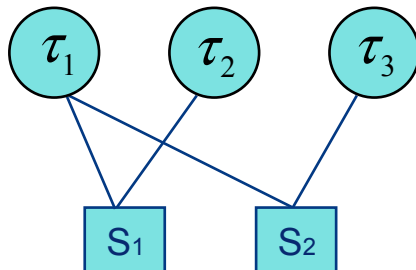
Determining the blocking factor for task  $\tau_i$ :

1. Determine the ceiling priorities for all critical regions.
2. Identify the tasks that have a priority lower than  $\tau_i$  and that calls critical regions with a ceiling priority equal to or higher than the priority of  $\tau_i$ .
3. Consider the times that these tasks lock the actual critical regions. The longest of those times constitutes the blocking factor  $B_i$ .

## Example 2: scheduling using DM

**Problem:** Assume a system with three tasks using two resources, according to the figure below. The timing properties of the tasks are given in the table. Note that  $D_i \leq T_i$ .

Two semaphores,  $S_1$  and  $S_2$ , are used for protecting the resources. The parameters  $H_{S_1}$  and  $H_{S_2}$  represent the longest time a task may lock semaphore  $S_1$  and  $S_2$ , respectively.



| Task     | $C_i$ | $D_i$ | $T_i$ | $H_{S_1}$ | $H_{S_2}$ |
|----------|-------|-------|-------|-----------|-----------|
| $\tau_1$ | 2     | 4     | 5     | 1         | 1         |
| $\tau_2$ | 3     | 12    | 12    | 1         | -         |
| $\tau_3$ | 8     | 24    | 25    | -         | 2         |

# Example 2: scheduling using DM

## Problem: (cont'd)

Examine the schedulability of the tasks when ICPP (Immediate Ceiling Priority Protocol) is used.

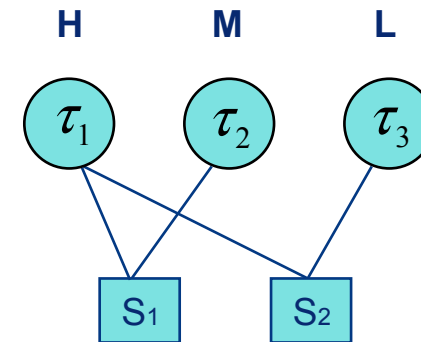
- a) Derive the ceiling priorities of the semaphores.
- b) Derive the blocking factors for the tasks.
- c) Determine whether the tasks are schedulable or not using DM.

## Example 2: scheduling using DM

a) Ceiling priorities for the semaphores:

$$S_1 = \max\{H, M\} = H$$

$$S_2 = \max\{H, L\} = H$$



b) Since both semaphores have highest ceiling priority (H), tasks  $\tau_1$  and  $\tau_2$  may be blocked by a task with lower priority regardless of which semaphore that lower-priority task uses.

$$B_1 = \max\{1, 2\} = 2$$

$\tau_2$  may use semaphore  $S_1$  or  
 $\tau_3$  may use semaphore  $S_2$

$$B_2 = \max\{2\} = 2$$

$\tau_3$  may use semaphore  $S_2$

$$B_3 = 0$$

NOTE:  $\tau_2$  may be blocked although it does not use  $S_2$

## Example 2: scheduling using DM

c) Calculate response times:

$$R_1 = C_1 + B_1 = 2 + 2 = 4 \leq D_1 = 4 \Rightarrow \text{OK!}$$

$$R_2 = C_2 + B_2 + \left\lceil \frac{R_2}{T_1} \right\rceil C_1 \quad [\text{Assume } R_2^0 = C_2 = 3]$$

$$R_2^1 = 3 + 2 + \left\lceil \frac{3}{5} \right\rceil \cdot 2 = 3 + 2 + 1 \cdot 2 = 7$$

$$R_2^2 = 3 + 2 + \left\lceil \frac{7}{5} \right\rceil \cdot 2 = 3 + 2 + 2 \cdot 2 = 9$$

$$R_2^3 = 3 + 2 + \left\lceil \frac{9}{5} \right\rceil \cdot 2 = 3 + 2 + 2 \cdot 2 = 9 \leq D_2 = 12 \Rightarrow \text{OK!}$$

## Example 2: scheduling using DM

$$R_3 = C_3 + \left\lceil \frac{R_3}{T_2} \right\rceil C_2 + \left\lceil \frac{R_3}{T_1} \right\rceil C_1 \quad [\text{Assume } R_3^0 = C_3 = 8]$$

$$R_3^1 = 8 + \left\lceil \frac{8}{12} \right\rceil \cdot 3 + \left\lceil \frac{8}{5} \right\rceil \cdot 2 = 8 + 1 \cdot 3 + 2 \cdot 2 = 15$$

$$R_3^2 = 8 + \left\lceil \frac{15}{12} \right\rceil \cdot 3 + \left\lceil \frac{15}{5} \right\rceil \cdot 2 = 8 + 2 \cdot 3 + 3 \cdot 2 = 20$$

$$R_3^3 = 8 + \left\lceil \frac{20}{12} \right\rceil \cdot 3 + \left\lceil \frac{20}{5} \right\rceil \cdot 2 = 8 + 2 \cdot 3 + 4 \cdot 2 = 22$$

$$R_3^4 = 8 + \left\lceil \frac{22}{12} \right\rceil \cdot 3 + \left\lceil \frac{22}{5} \right\rceil \cdot 2 = 8 + 2 \cdot 3 + 5 \cdot 2 = 24$$

$$R_3^5 = 8 + \left\lceil \frac{24}{12} \right\rceil \cdot 3 + \left\lceil \frac{24}{5} \right\rceil \cdot 2 = 8 + 2 \cdot 3 + 5 \cdot 2 = 24 \leq D_3 = 24 \Rightarrow \text{OK!}$$