Facultat d'Informàtica de Barcelona
Universitat Politècnica de Catalunya

Real-Time Systems

3b-Rate Monotonic

Antonio Camacho Santiago
antonio.camacho.santiago@upc.edu

Objective

To understand rate monotonic (RM) schedulers…

To design rate monotonic schedulers…

To analyze rate monotonic …

To evaluate the pros and cons of rate monotonic schedulers…

```
at design stage:
  assign higher priorities to shorter period tasks

at runtime each Sys_Tick:
  for each active task
    dispatch the task with higher priority
```

Rate Monotonic scheduler

Durign the design of the system, each task has a priority proportional to its rate. Higher request rates will have higher priorities

$$\forall \tau_i, \tau_j : T_i < T_j \Rightarrow P_i > P_j$$

It can also be used the following rule saying that priorities are assigned proportionally to the inverse of the period

$$P_i \propto \frac{1}{T_i}$$

At each system tick, the scheduler looks for the existing active tasks to dispatch the task with higher priority. Thus, preemption is allowed at each system tick

More details can be found in C. L. Liu and J. W. Layland, “Scheduling Algorithms for Multiprogramming in a Hard-Real-Time Environment,” Journal of the Association for Computing Machinery, vol. 20, n. 1, Jan. 1973, pp. 46-61

Requisites

The first approach for the rate monotonic scheduler is based on periodic tasks as follows:

- 1 microprocessor
- Static tasks
- Periodic tasks
- No precedence among tasks
- The WCET for each task is known, fitted and less than its deadline
- Deadlines of each task are equal to their periods
- **Tasks can be preempted**
- **RT kernel uses fixed priorities**

The schedulability analysis tries to know in advance if all the release times for each task occurs before its deadline.

The analysis is performed at the critical time (not during the whole hiperperiod): for a system of periodic independent tasks scheduled with fixed priorities, each instant in which a task is activated at the same time that each one of the higher priority tasks is called a critical time

Methodology

3b-Rate Monotonic

Assign priorities based on inverse of periods

Task τ_i	Computing time c_i ms)	Period T_i = Deadline D_i (ms)	Priority
τ_1	3	7	3
τ_2	3	12	2
τ_3	5	20	1

Check one of the following two feasibility sufficient conditions (only applies to $D_i = T_i$ and no shared resources)
Sufficient condition 1:

$$U_{\text{total}} = \sum_{i=1}^{n} U_i = \sum_{i=1}^{n} \frac{c_i}{T_i} = \frac{c_1}{T_1} + \frac{c_2}{T_2} + \cdots + \frac{c_n}{T_n} \leq n\left(2^{1/n} - 1\right)$$

Sufficient condition 2: If condition 1 is not fulfilled, check hyperbolic bound condition.

$$\prod_{i=1}^{n} (U_i + 1) \leq 2$$

Once condition 1 or 2 are fulfilled, you've done
If neither total utilization nor hyperbolic bound is demonstrated, check response time analysis for the critical time (necessary and sufficient condition). It represents the **interference** due to preemption of higher priority tasks. This condition applies to $D_i \leq T_i$

$$\forall \tau_i : R_i = C_i + \sum_{j \in hp(i)} \left\lceil \frac{R_i}{T_j} \right\rceil C_j \leq D_i \qquad hp(i) = \{j : 1..n \mid P_j > P_i\}$$

Methodology: Response Time Analysis 3b-Rate Monotonic

Response Time Analysis is used to check the feasibility of the task scheduling
It presents a necessary and sufficient condition
The tests applies to $D_i \leq T_i$
It is based on the response time due to the computing time C_i plus the interference I_i produced by higher priority tasks preempting lower priority tasks

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

The solution is obtained recurrently based on *n* iteration to compute *n+1*

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

The recurrence starts with $W_1^0 = C_1$ corresponding to the response time of task 1 which cannot be preempted by any other task
It finishes when $W_i^{n+1} = W_i^n$ which implies that the solution is in steady-state and the response time of task i is $R_i = W_i^n$.
It can also finish whenever the release time is greater than the deadline for any task (unfeasible)

Example 1

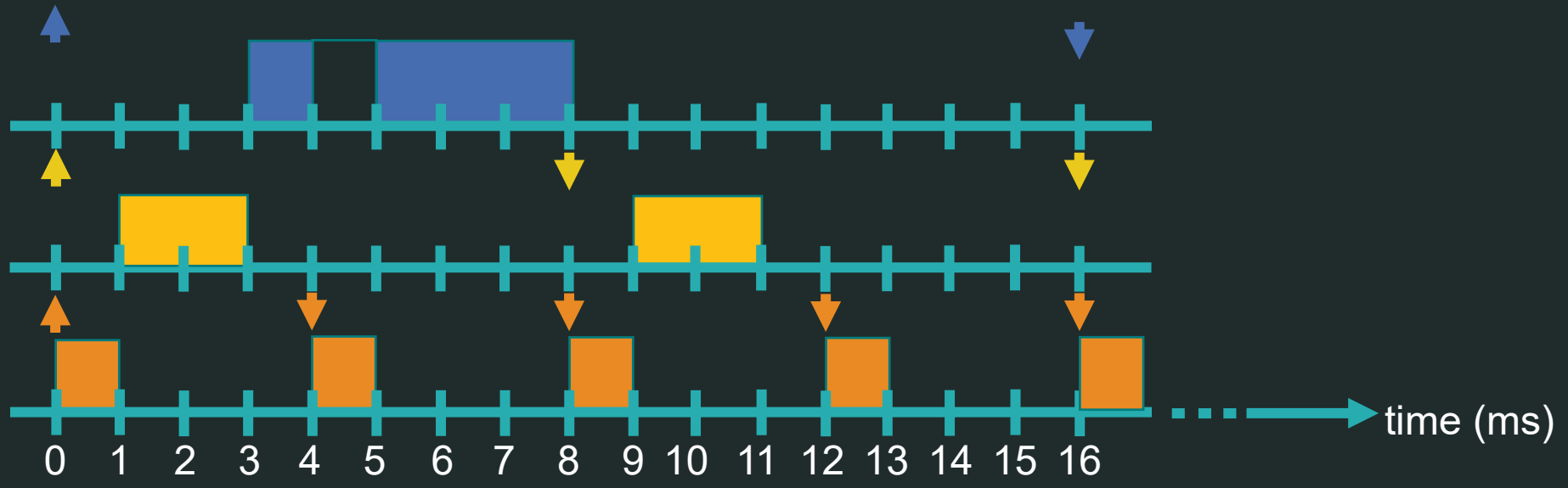
Check sufficient conditions

$$U_{\text{total}} = \sum_{i=1}^{n} U_i = \sum_{i=1}^{n} \frac{c_i}{T_i} \leq n\left(2^{1/n} - 1\right)$$

$$U_{\text{total}} = \frac{1}{4} + \frac{2}{8} + \frac{4}{16} = 0.75 \leq 0.78$$

$$\prod_{i=1}^{n} (U_i + 1) = \left(\frac{1}{4} + 1\right)\left(\frac{2}{8} + 1\right)\left(\frac{4}{16} + 1\right) = 1.25 \cdot 1.25 \cdot 1.25 = 1.95 \leq 2$$

Task τ_i	Computing time c_i (ms)	Period T_i = Deadline D_i (ms)	Priority
τ_1	1	4	3
τ_2	2	8	2
τ_3	4	16	1



Example 1

Response Time Analysis (not required since $n(2^{1/n}-1)$ condition and hyperbolic are fulfilled)

Task τ_i	Computing time c_i (ms)	Period T_i = Deadline D_i (ms)	Priority
τ_1	1	4	3
τ_2	2	8	2
τ_3	4	16	1

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

$i = 1$: $W_1^0 = C_1 + 0 = 1 \leq 4 = D_i$

$i = 2$: $W_2^0 = C_2 + 0 = 2 \leq 8 = D_i$

$W_2^1 = C_2 + \left\lceil \frac{2}{4} \right\rceil 1 = 2 + 1 = 3 \leq 8 = D_i$

$W_2^2 = C_2 + \left\lceil \frac{3}{4} \right\rceil 1 = 2 + 1 = 3 \leq 8 = D_i$

$W_2^2 = W_2^1 \rightarrow R_2 = 3$

$i = 3$: $W_3^0 = C_3 + 0 + 0 = 4 \leq 16 = D_i$

$W_3^1 = C_3 + \left\lceil \frac{4}{4} \right\rceil 1 + \left\lceil \frac{4}{8} \right\rceil 2 = 4 + 1 \cdot 1 + 1 \cdot 2 = 7 \leq 16 = D_i$

$W_3^2 = C_3 + \left\lceil \frac{7}{4} \right\rceil 1 + \left\lceil \frac{7}{8} \right\rceil 2 = 4 + 2 \cdot 1 + 1 \cdot 2 = 8 \leq 16 = D_i$

$W_3^3 = C_3 + \left\lceil \frac{8}{4} \right\rceil 1 + \left\lceil \frac{8}{8} \right\rceil 2 = 4 + 2 \cdot 1 + 1 \cdot 2 = 8 \leq 16 = D_i$

$W_3^5 = W_3^4 \rightarrow R_3 = 8$

Hyperperiod has not been checked, just the critical time instant in which all tasks have been activated at the same time

Example 2

Check utilization

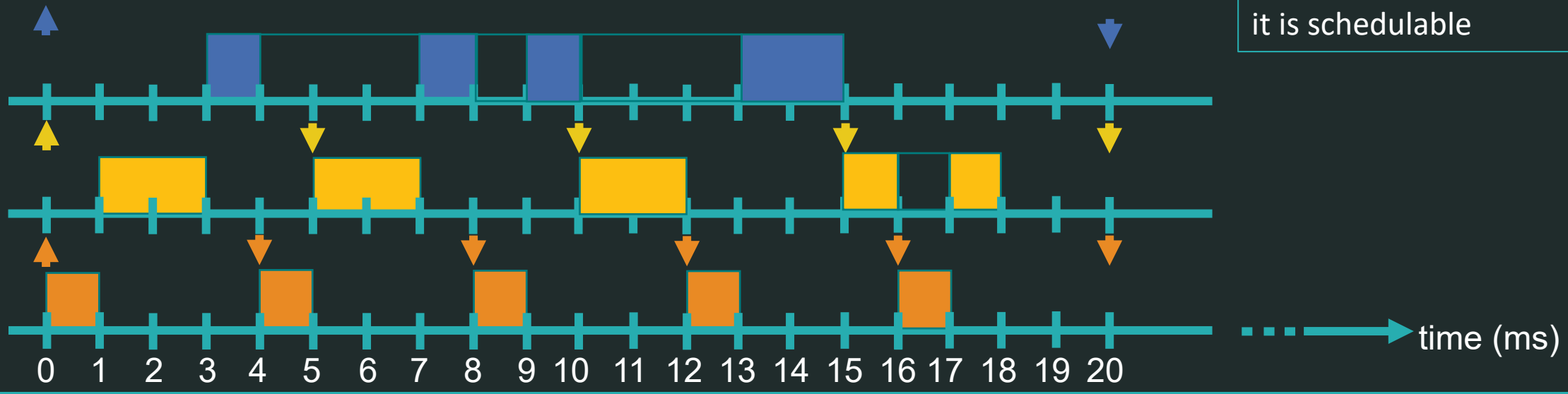
$$U_{\text{total}} = \sum_{i=1}^{n} U_i = \sum_{i=1}^{n} \frac{c_i}{T_i} \leq n\left(2^{1/n} - 1\right)$$

$$U_{\text{total}} = \frac{1}{4} + \frac{2}{5} + \frac{5}{20} = 0.90 \geq \mathbf{0.78}$$

$$\prod_{i=1}^{n} (U_i + 1) = \left(\frac{1}{4} + 1\right)\left(\frac{2}{5} + 1\right)\left(\frac{5}{20} + 1\right) = 1.25 \cdot 1.4 \cdot 1.25 = 2.18 \geq 2$$

Task τ_i	Computing time c_i (ms)	Period T_i = Deadline D_i (ms)	Priority
τ_1	1	4	3
τ_2	2	5	2
τ_3	5	20	1

Sufficient conditions do not hold! However, it is schedulable



Example 2

Response Time Analysis

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

$i = 1$: $W_1^0 = C_1 + 0 = 1 \leq 4 = D_i$

$i = 2$: $W_2^0 = C_2 + 0 = 2 \leq 5 = D_i$

$W_2^1 = C_2 + \left\lceil \frac{2}{4} \right\rceil 1 = 2 + 1 = 3 \leq 5 = D_i$

$W_2^2 = C_2 + \left\lceil \frac{3}{4} \right\rceil 1 = 2 + 1 = 3 \leq 5 = D_i$

$W_2^2 = W_2^1 \rightarrow R_2 = 3$

$i = 3$: $W_3^0 = C_3 + 0 + 0 = 5 \leq 5 = D_i$

$W_3^1 = C_3 + \left\lceil \frac{5}{4} \right\rceil 1 + \left\lceil \frac{5}{5} \right\rceil 2 = 5 + 2 \cdot 1 + 1 \cdot 2 = 9 \leq 20 = D_i$

$W_3^2 = C_3 + \left\lceil \frac{9}{4} \right\rceil 1 + \left\lceil \frac{9}{5} \right\rceil 2 = 5 + 3 \cdot 1 + 2 \cdot 2 = 12 \leq 20 = D_i$

$W_3^3 = C_3 + \left\lceil \frac{12}{4} \right\rceil 1 + \left\lceil \frac{12}{5} \right\rceil 2 = 5 + 3 \cdot 1 + 3 \cdot 2 = 14 \leq 20 = D_i$

$W_3^4 = C_3 + \left\lceil \frac{14}{4} \right\rceil 1 + \left\lceil \frac{14}{5} \right\rceil 2 = 5 + 4 \cdot 1 + 3 \cdot 2 = 15 \leq 20 = D_i$

$W_3^5 = C_3 + \left\lceil \frac{15}{4} \right\rceil 1 + \left\lceil \frac{15}{5} \right\rceil 2 = 5 + 4 \cdot 1 + 3 \cdot 2 = 15 \leq 20 = D_i$

$W_3^5 = W_3^4 \rightarrow R_3 = 15$

Task τ_i	Computing time c_i (ms)	Period T_i = Deadline D_i (ms)	Priority
τ_1	1	4	3
τ_2	2	5	2
τ_3	5	20	1

Example 3

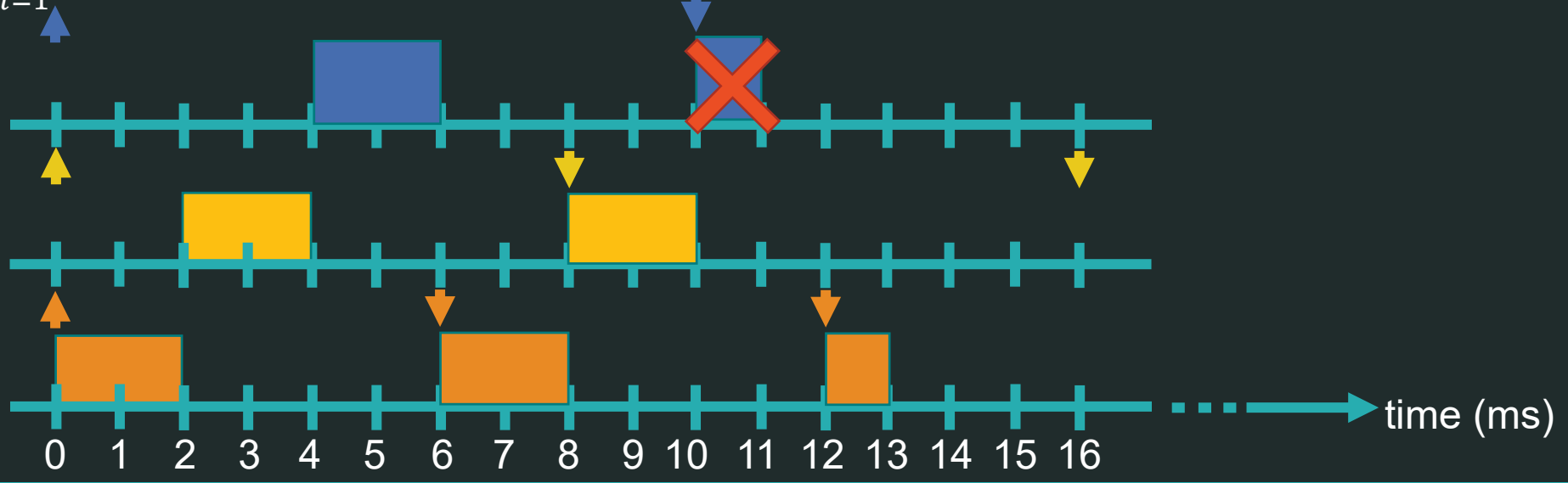
Check utilization

$$U_{\text{total}} = \sum_{i=1}^{n} U_i = \sum_{i=1}^{n} \frac{c_i}{T_i} \leq n\left(2^{1/n} - 1\right)$$

$$U_{\text{total}} = \frac{2}{6} + \frac{2}{8} + \frac{3}{10} = 0.88 \geq \mathbf{0.78}$$

$$\prod_{i=1}^{n}(U_i + 1) = \left(\frac{2}{6} + 1\right)\left(\frac{2}{8} + 1\right)\left(\frac{3}{10} + 1\right) = 1.33 \cdot 1.25 \cdot 1.3 = 2.16 \geq 2$$

Task τ_i	Computing time c_i (ms)	Period T_i = Deadline D_i (ms)	Priority
τ_1	2	6	3
τ_2	2	8	2
τ_3	3	10	1



Example 3

Response Time Analysis

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

Task τ_i	Computing time c_i (ms)	Period T_i = Deadline D_i (ms)	Priority
τ_1	2	6	3
τ_2	2	8	2
τ_3	3	10	1

$i = 1$: $W_1^0 = C_1 + 0 = 2 \leq 6 = D_i$

$i = 2$: $W_2^0 = C_2 + 0 = 2 \leq 8 = D_i$

$W_2^1 = C_2 + \left\lceil \frac{2}{6} \right\rceil 2 = 2 + 1 \cdot 2 = 3 \leq 8 = D_i$

$W_2^2 = C_2 + \left\lceil \frac{2}{6} \right\rceil 2 = 2 + 1 \cdot 2 = 3 \leq 8 = D_i$

$W_2^2 = W_2^1 \rightarrow R_2 = 3$

$i = 3$: $W_3^0 = C_3 + 0 + 0 = 3 \leq 10 = D_i$

$W_3^1 = C_3 + \left\lceil \frac{3}{6} \right\rceil 2 + \left\lceil \frac{3}{8} \right\rceil 2 = 3 + 2 \cdot 2 + 2 \cdot 2 = 11 \geq 10 = D_i$

Rate Monotonic scheduler

Pros:

The rate monotonic scheduler is based on fixed priorities configured at design stage according to the rate of occurrence of each task

Sufficient condition $U_{\text{total}} \leq n\left(2^{1/n} - 1\right)$ or hyperbolic bound $\prod_{i=1}^{n}(U_i + 1) \leq 2$

Necessary and sufficient condition $\forall \tau_i: R_i = C_i + \sum_{j \in hp(i)} \left\lceil \frac{R_i}{T_j} \right\rceil C_j \leq D_i$

Feasibility of scheduler is performed at the critical time

Optimality: among all the fixed priorities policies, RM is optimal, i.e. if some priority assignment ensures schedulability, then RM will also ensure it (converse is not truth)

Cons:

Preemption

Performance depends on the system tick

Very low CPU utilization to guarantee the deadlines. On the limit when $n \to \infty$

$$\lim_{n\to\infty} U_{\text{total}} = \lim_{n\to\infty} n\left(2^{1/n} - 1\right) = \ln(2) = 0.6931$$