THE UNIVERSITY OF KANSAS

Lecture 19: Real-Time Process Scheduling – Schedulability

EECS 388 – Fall 2022

Lecture notes are based on slides created by Prof. Heechul Yun

So far

- Job, Task
- Periodic task model
  - ti = (Ci, Pi) or (Ci, Pi, Di)
- Static/dynamic priority scheduling:
  - RM
  - EDF
- Utilization
  - Ui = Ci / Pi

$$U = \sum_i \frac{C_i}{p_i}$$

Outline

Is there a way to check whether an RM schedule is feasible?

- Utilization Bound
- Exact Schedulability Analysis

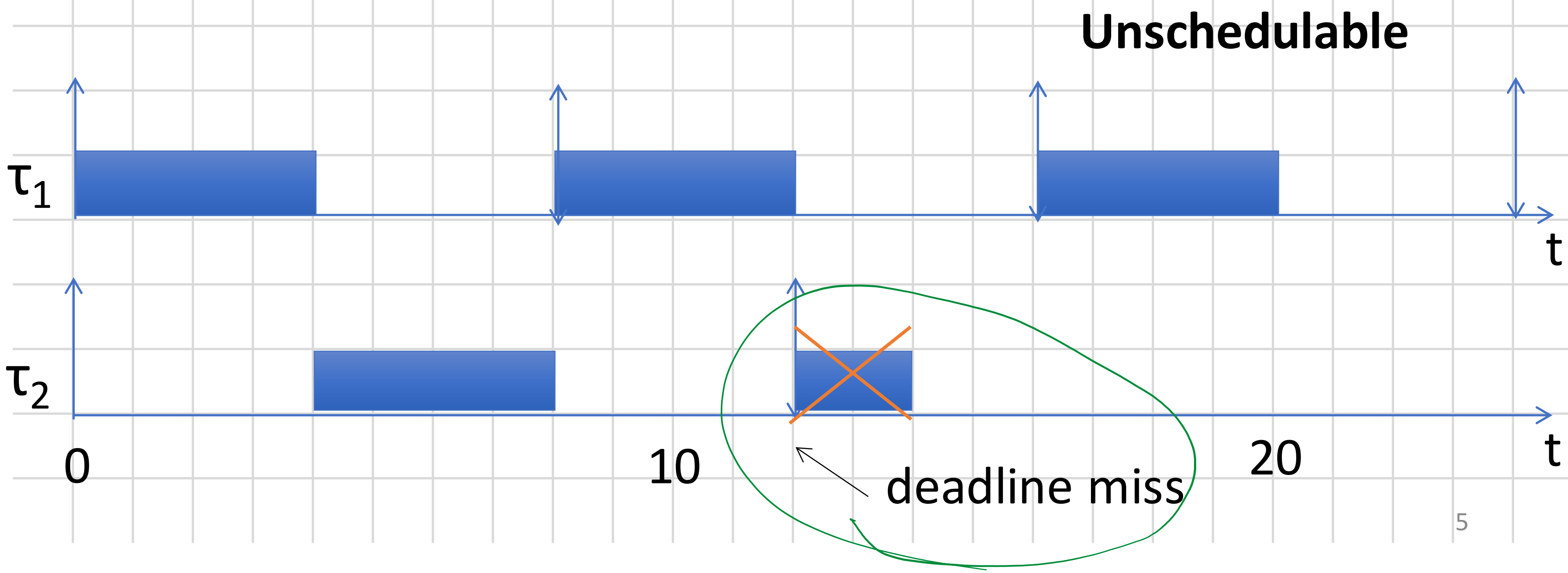
Outline

Is there a way to check whether an RM schedule is feasible?

- Utilization Bound
- Exact Schedulability Analysis

Recall: RM Example

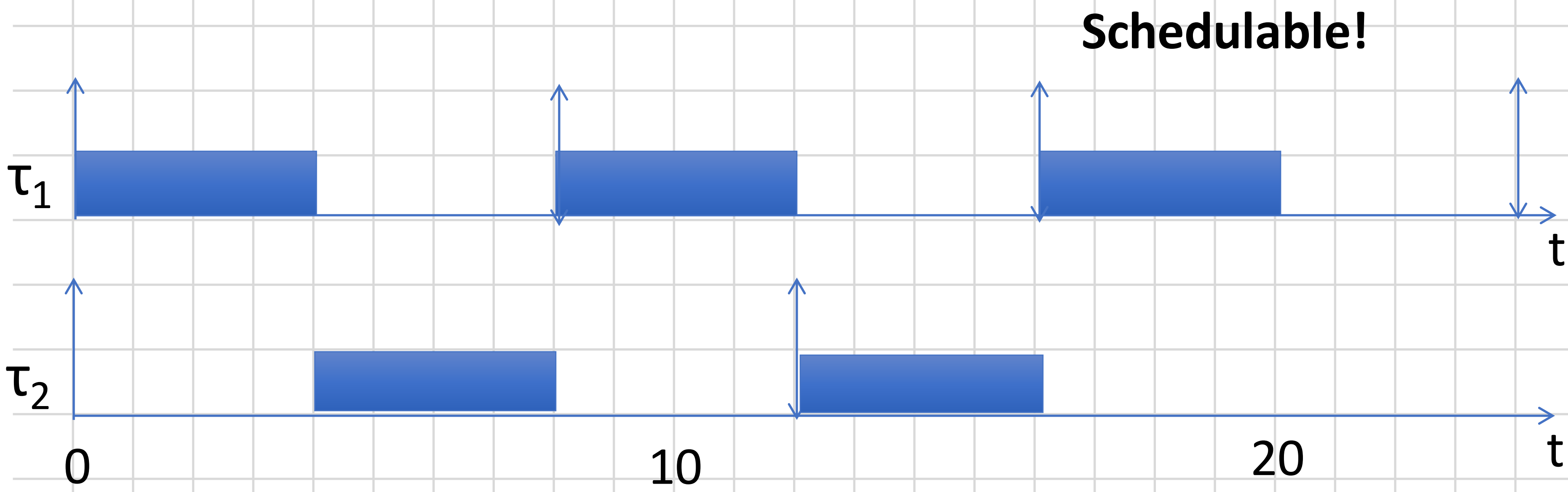
- τ_1 (C1 = 4, T1 = 8), high prio
- τ_2 (C2 = 6, T1 = 12), low prio
- Utilization: U = 4/8 + 6/12 = 1



Is there an easy way to know whether a taskset is schedulable or not?

RM Example

- τ_1 (C1 = 4, T1 = 8), high prio
- τ_2 (C2 = **4**, T1 = 12), low prio
- Utilization: U = 4/8 + **4**/12 = 10/12 = 0.83



Liu & Layland, JACM, Jan. 1973

Scheduling Algorithms for Multiprogramming in a Hard-Real-Time Environment

C. L. LIU

Project MAC, Massachusetts Institute of Technology

AND

JAMES W. LAYLAND

Jet Propulsion Laboratory, California Institute of Technology

ABSTRACT. The problem of multiprogram scheduling on a single processor is studied from the viewpoint of the characteristics peculiar to the program functions that need guaranteed service. It is shown that an optimum fixed priority scheduler possesses an upper bound to processor utilization which may be as low as 70 percent for large task sets. It is also shown that full processor utilization can be achieved by dynamically assigning priorities on the basis of their current deadlines. A combination of these two scheduling techniques is also discussed.

KEY WORDS AND PHRASES: real-time multiprogramming, scheduling, multiprogram scheduling, dynamic scheduling, priority assignment, processor utilization, deadline driven scheduling

CR CATEGORIES: 3.80, 3.82, 3.83, 4.32

Liu & Layland Bound

- A set of n periodic tasks is schedulable if

$$\boxed{\frac{c_1}{p_1} + \frac{c_2}{p_2} + \ldots + \frac{c_n}{p_n}} \leq \boxed{n(2^{1/n} - 1)}$$

total util UB

- UB(1) = 1.0
- UB(2) = 0.828
- UB(3) = 0.779
- …
- UB(n) where n is large?

$$\boxed{\lim_{n\to\infty} n(2^{1/n} - 1) = \ln(2) \approx 0.693.}$$

Q. If it isn't, does that mean the taskset is unschedulable?

A. Not necessarily.
It's a sufficient condition,
but not necessary one.

Sample Problem

	C	T	U
Task $\tau 1$	20 40	100	0.200
Task $\tau 2$	40	150	0.267
Task $\tau 3$	100	350	0.286

L&L Bound

UB(1) = 1.0
UB(2) = 0.828
$\boxed{\text{UB(3) = 0.779}}$
UB(n) = 0.693

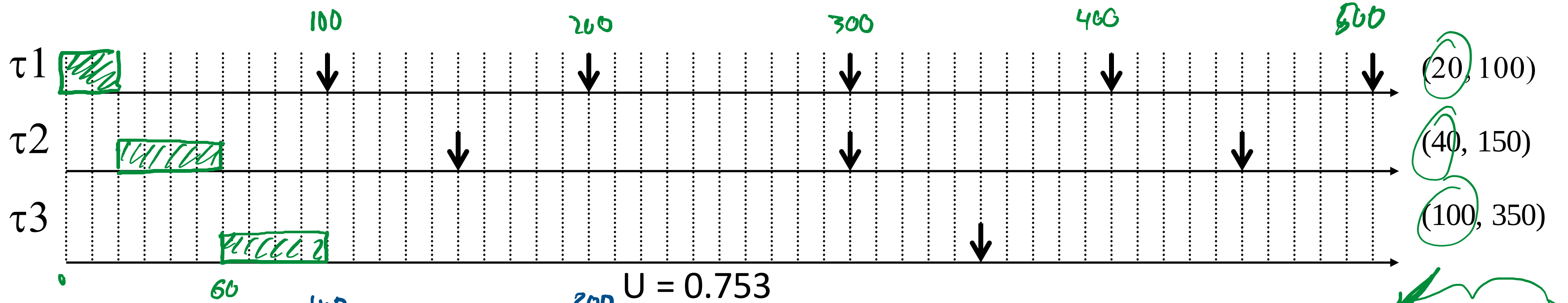
- Are all tasks schedulable?
  - $U_1 + U_2 + U_3 = 0.753 < U(3)$ ➔ **Schedulable!**
- What if we double the C of $\tau 1$
  - $0.2*2 + 0.267 + 0.286 = \boxed{0.953} > UB(3) = 0.779$
  - **We don't know yet.**

Sample Problem

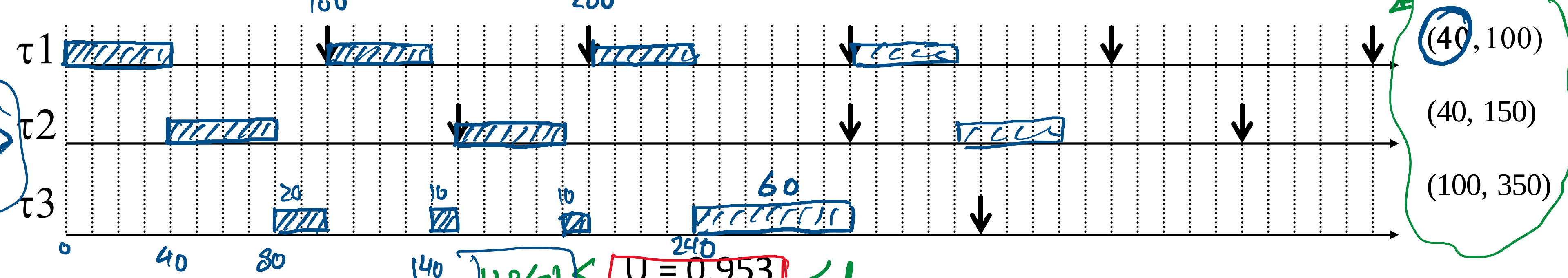
L&L Bound

UB(3) = 0.779

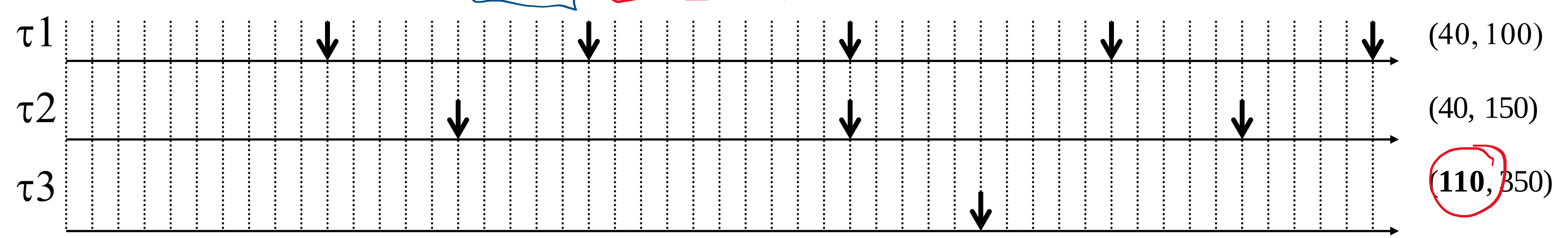
UB(n) = 0.693



U = 0.753



UB(3) < U = 0.953 ≤ 1

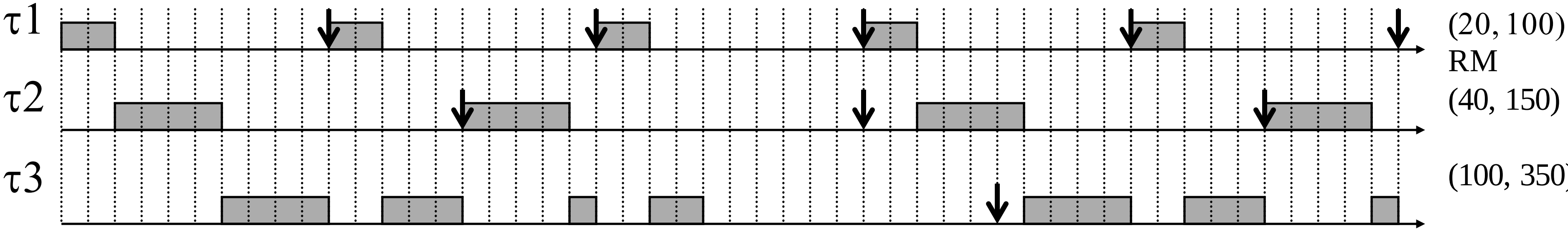


U = 0.981

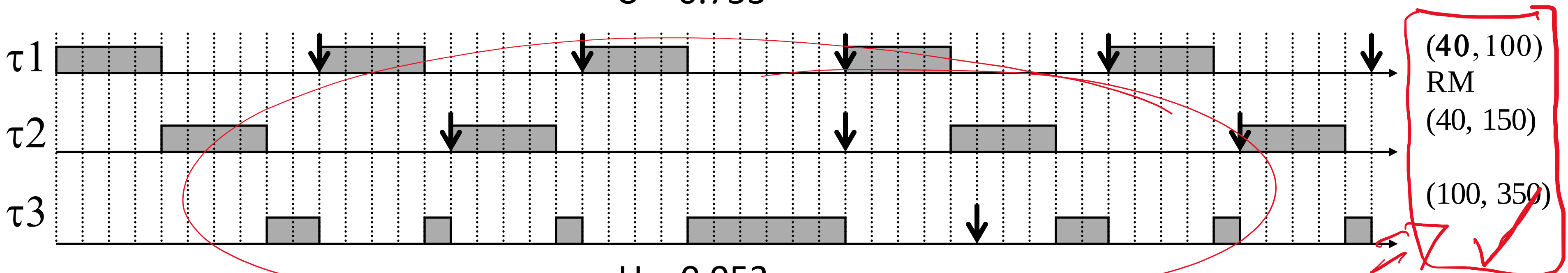
Sample Problem

L&L Bound

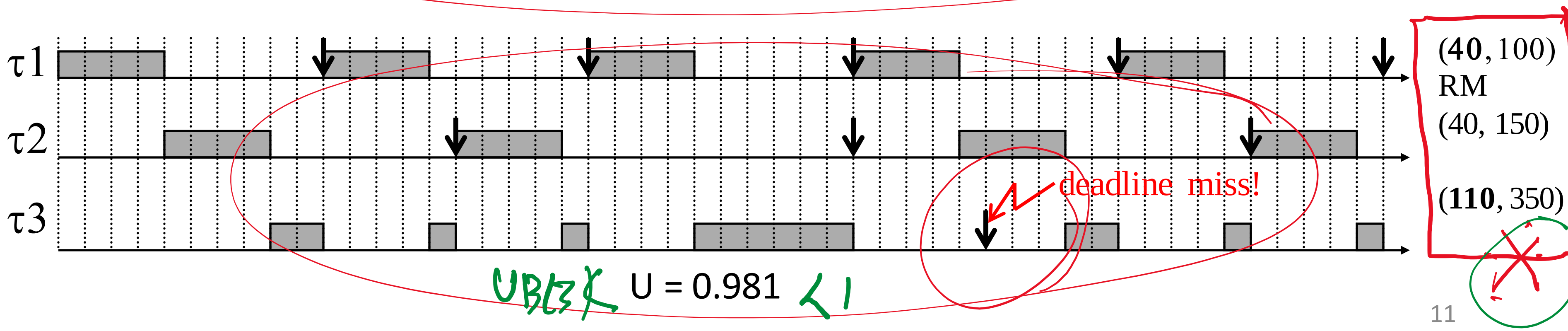
UB(3) = 0.779
UB(n) = 0.693



U = 0.753



U = 0.953



UB(3) < U = 0.981 < 1

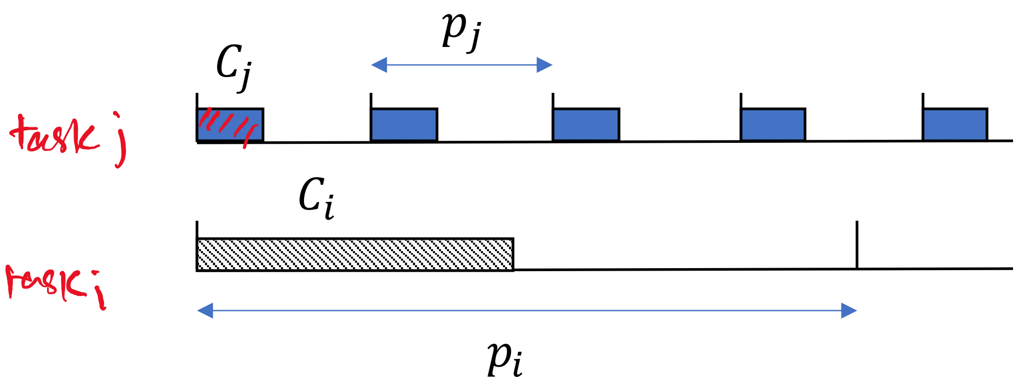
Outline

Is there a way to check whether an RM schedule is feasible?

- Utilization Bound
- Exact Schedulability Analysis

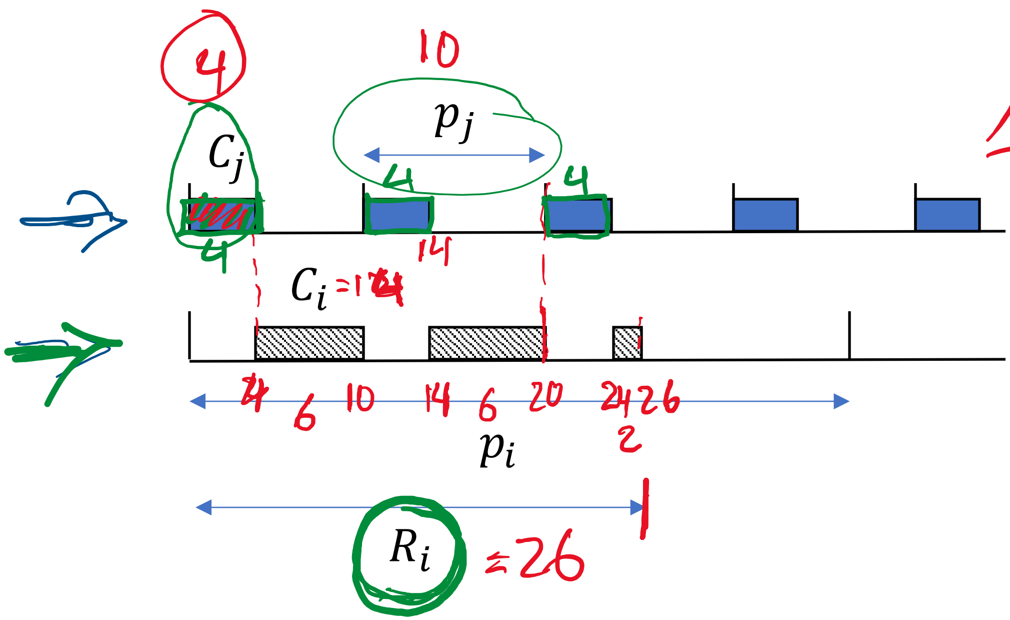
Exact Schedulability Test

- Interference from higher priority task j ?



Exact Schedulability Test

- Interference from higher priority task j ?



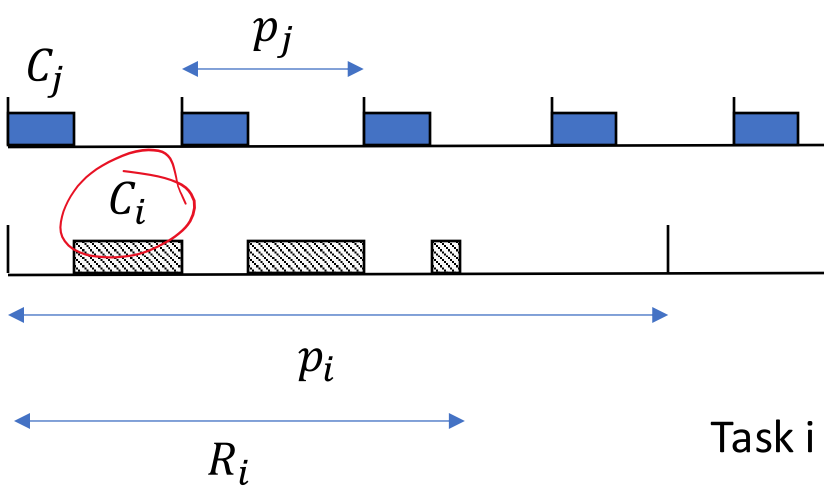
Inter => $\left\lceil \frac{Ri}{Pj} \right\rceil C_j$

$\left\lceil \frac{26}{10} \right\rceil = 3$

Interference = 3 × 4 => 12

Exact Schedulability Test

- Interference from higher priority task j ?



$$\left\lceil \frac{Ri}{Pj} \right\rceil C_j$$

Interference from higher priority tasks

Task i execution time

Schedulability condition for first iteration of task i:

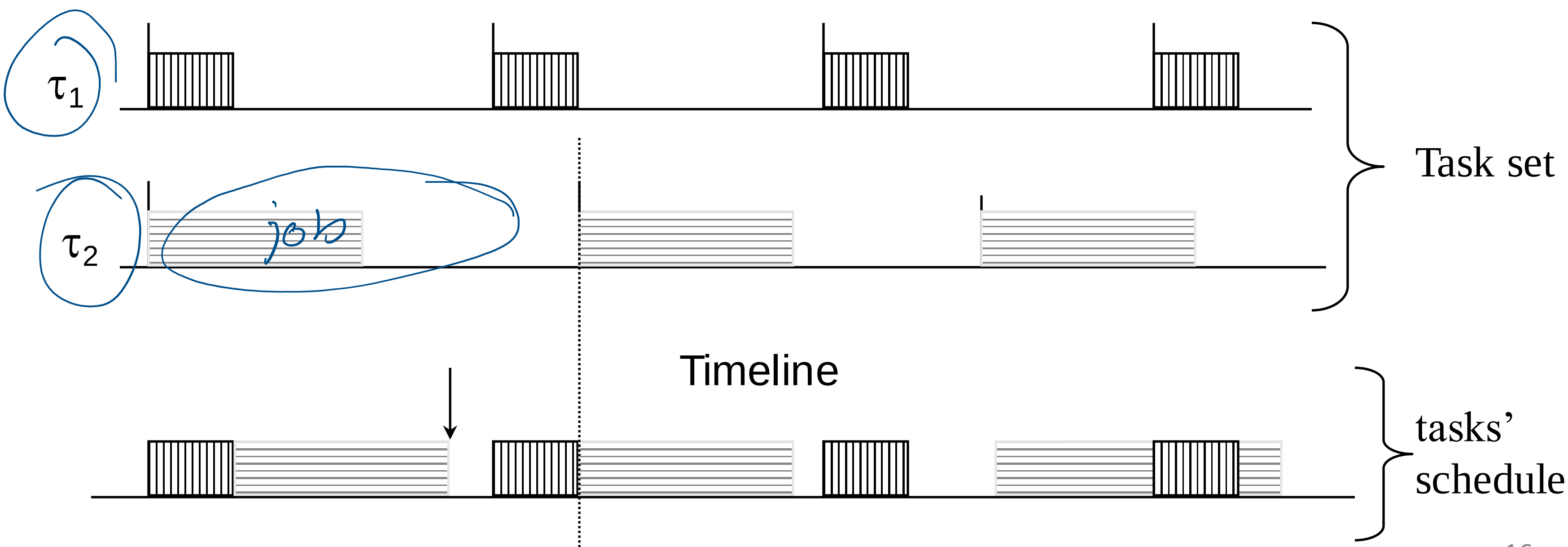
$$C_i + \underbrace{\sum_{j=0}^{i} \left\lceil \frac{Ri}{Pj} \right\rceil C_j}_{+\ \text{interference}} \leq p_i$$

Deadline

$R_{(i)}$

Critical Instant Theorem

- If a task meets its first deadline when all higher priority tasks are started at the same time, then this task's future deadlines will always be met.



RMS

Exact Schedulability Test

- For each task, checks if it can meet its first deadline

ceiling function

$$r_i^{k+1} = c_i + \sum_{j=1}^{i-1} \left\lceil \frac{r_i^k}{p_j} \right\rceil c_j, \quad \text{where } r_i^0 = \sum_{j=1}^{i} c_j$$

task

Test terminates when $r_i^{k+1} > p_i$ (not schedulable)
or when $r_i^{k+1} = r_i^k \le p_i$ (schedulable).

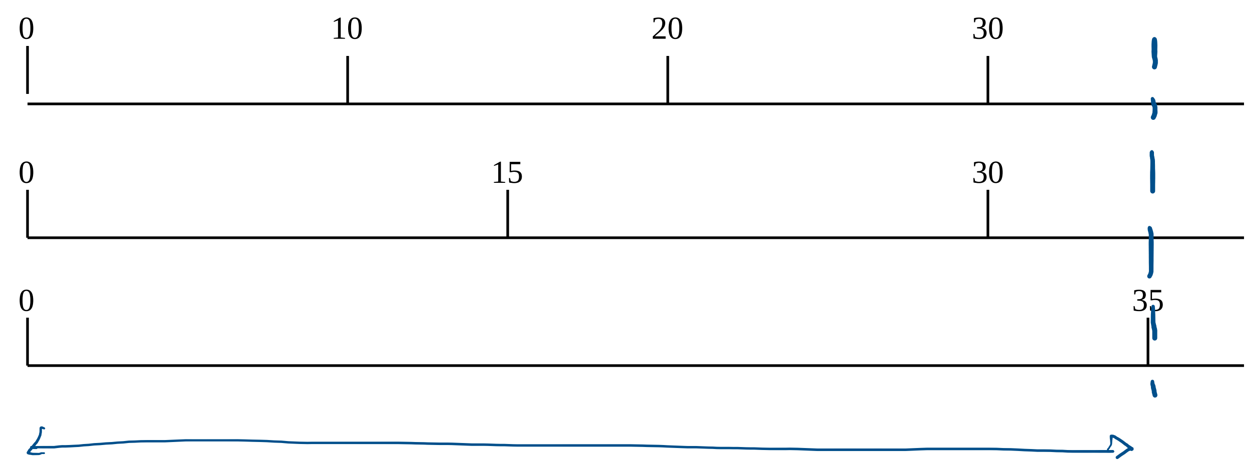
Exact Schedulability Test

- Following taskset $\boxed{r_1 \leq 10}$, $\boxed{r_2 \leq 15}$, $\boxed{r_3 \leq 35}$ ✓

ask1 $(c_1 = 4, p_1 = 10), U_1 = 0.4$

ask2 $(c_2 = 4, p_2 = 15), U_2 = 0.27$

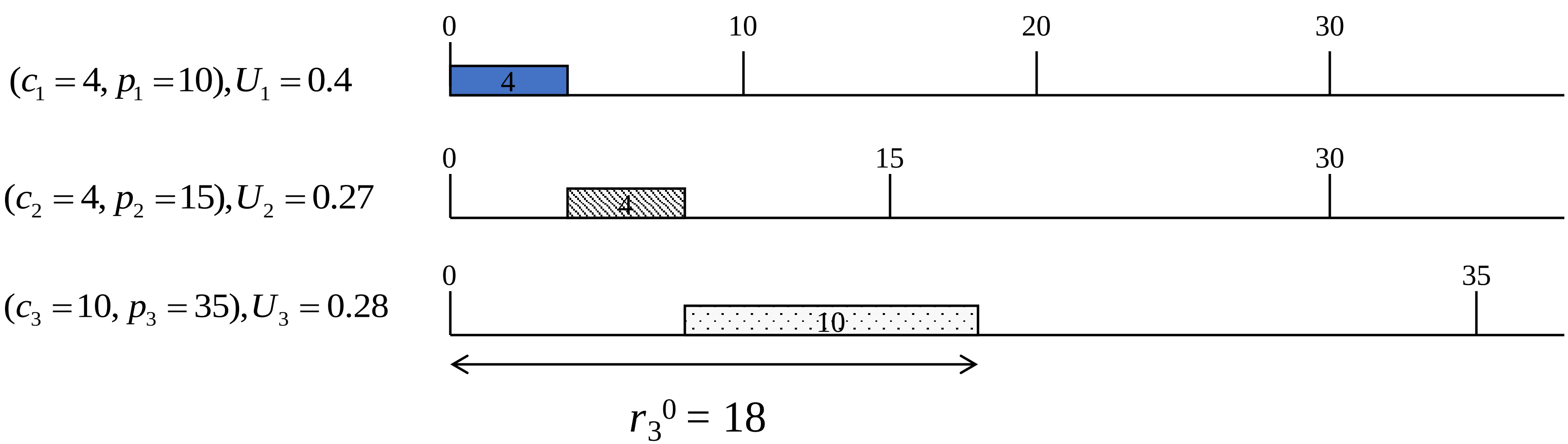
sk3 $(c_3 = 10, p_3 = 35), U_3 = 0.28$



Exact Schedulability Test

- For task 3
  - First iteration

$$r_3^0 = \sum_{j=1}^{3} c_j = c_1 + c_2 + c_3 = 4 + 4 + 10 = 18$$

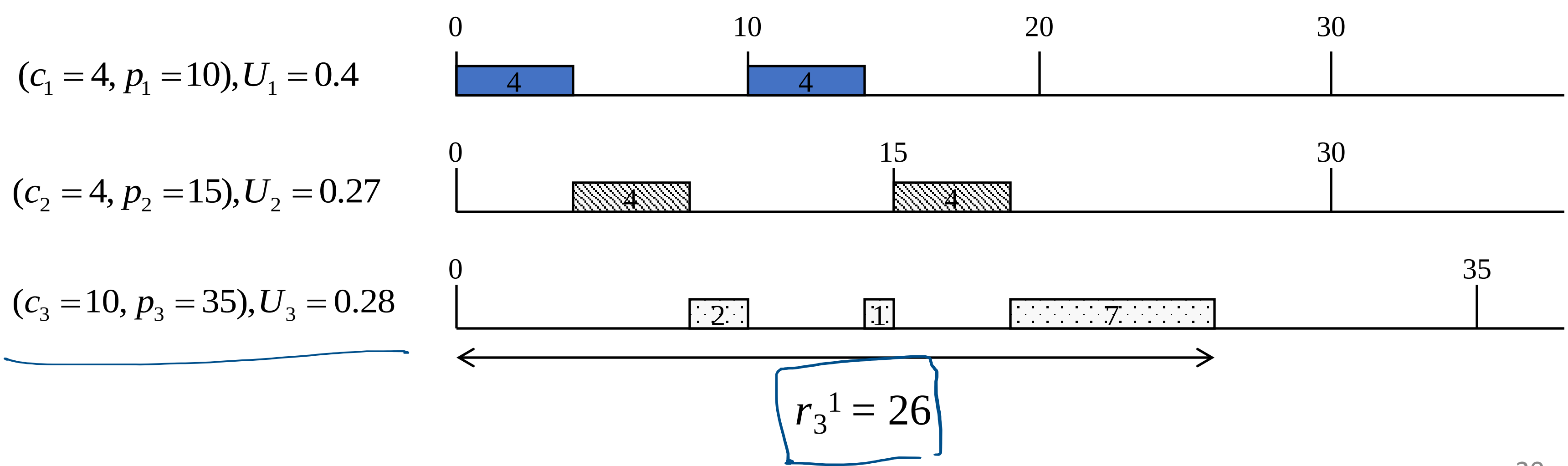


Exact Schedulability Test

- For task 3
  - Second iteration

task 1 task 2

$$r_3^1 = c_3 + \sum_{j=1}^{2} \left\lceil \frac{r_3^0}{p_j} \right\rceil \cdot c_j = 10 + \overset{2}{\cancel{\left\lceil \frac{18}{10} \right\rceil}} 4 + \overset{2}{\cancel{\left\lceil \frac{18}{15} \right\rceil}} 4 = 26$$



Exact Schedulability Test

- For task 3
  - Third iteration

task1 task2

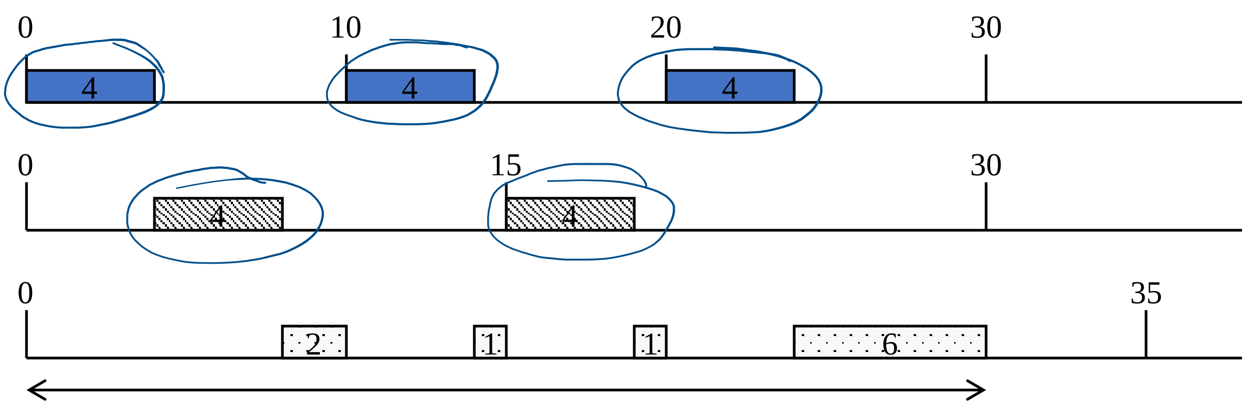
$$r_3^2 = c_3 + \sum_{j=1}^{2} \left\lceil \frac{r_3^1}{p_j} \right\rceil \cdot c_j = 10 + \left\lceil \frac{26}{10} \right\rceil 4 + \left\lceil \frac{26}{15} \right\rceil 4 = 30$$

2x4

$(c_1 = 4, p_1 = 10), U_1 = 0.4$

$(c_2 = 4, p_2 = 15), U_2 = 0.27$

$(c_3 = 10, p_3 = 35), U_3 = 0.28$



$r_3{}^2 = r_3{}^3 = 30$

Exact Schedulability Test

- For task 3
  - Fourth iteration … is the same as the 3rd

$$r_3^3 = c_3 + \sum_{j=1}^{2} \left\lceil \frac{r_3^2}{p_j} \right\rceil \cdot c_j = \underbrace{10 + \left\lceil \frac{30}{10} \right\rceil 4 + \left\lceil \frac{30}{15} \right\rceil 4 = 30}$$

$$r_3^2 = c_3 + \sum_{j=1}^{2} \left\lceil \frac{r_3^1}{p_j} \right\rceil \cdot c_j = 10 + \left\lceil \frac{26}{10} \right\rceil 4 + \left\lceil \frac{26}{15} \right\rceil 4 = 30 \qquad \leq 35$$

- **Done!**

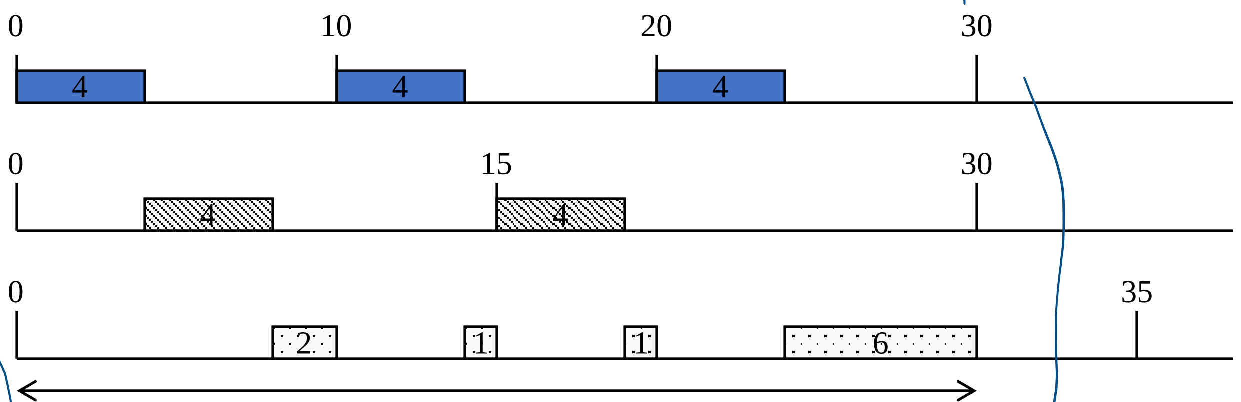
Exact Schedulability Test

- All tasks meet their deadlines → schedulable

$(c_1 = 4, p_1 = 10), U_1 = 0.4$

$(c_2 = 4, p_2 = 15), U_2 = 0.27$

$(c_3 = 10, p_3 = 35), U_3 = 0.28$



$r_3^2 = r_3^3 = 30$

RMS

Example

- Consider the following real-time taskset.

$r_1 \le 10$

Task	C	P	U
t1	4	10	0.400
t2	6	15	0.267
t3	10	35	0.286

0.953

- Is this taskset schedulable under the rate monotonic scheduling? Use the exact analysis for your answer.

$r_1^0 = 4$

$r_1^1 = 4 \le 10$ ✓

$r_2^0 = 4 + 6 = 10$

$r_2^1 = 6 + \underbrace{\sum_{j=1}^{1} \left\lceil \frac{10}{P_j} \right\rceil * C_j} = 6 + \left\lceil \cancel{\frac{10}{10}}^{1} \right\rceil \times 4 = 10 \le$

Task	C	P	U
t1	4	10	0.400
t2	6	15	0.267
t3	10	35	0.286

$$r_3^0 = 4 + 6 + 10 = 20$$

$$r_3^1 = C_3 + \sum_{j=1}^{2} \left\lceil \frac{r_3^0}{P_j} \right\rceil * C_j = 10 + \overset{2}{\left\lceil \frac{20}{10} \right\rceil} * 4 + \overset{2}{\left\lceil \frac{20}{15} \right\rceil} * 6$$

$$10 + 8 + 12 = 30$$

$$r_3^2 = C_3 + \sum_{j=1}^{2} \left\lceil \frac{r_3^1}{P_j} \right\rceil * C_j = 10 + \overset{3}{\left\lceil \frac{30}{10} \right\rceil} * 4 + \overset{2}{\left\lceil \frac{30}{15} \right\rceil} * 6$$

$$10 + 12 + 12 = \underline{34}$$

$$r_3^3 = 10 + \overset{4}{\left\lceil \frac{34}{10} \right\rceil} * 4 + \overset{3}{\left\lceil \frac{34}{15} \right\rceil} * 6 =$$

$$10 + 16 + 18 = 44 > 35$$

Assumptions

- So far the theories assume
  - All the tasks are periodic
  - Tasks are scheduled according to RMS
  - All tasks are independent and do not share resources (data)
  - Tasks do not self-suspend during their execution
  - Scheduler overhead (context-switch) is negligible

Acknowledgements

- These slides draw on materials developed by
  - Lui Sha and Marco Caccamo (UIUC)
  - Rodolfo Pellizzoni (U. Waterloo)
  - Edward A. Lee and Prabal Dutta (UCB) for EECS149/249A