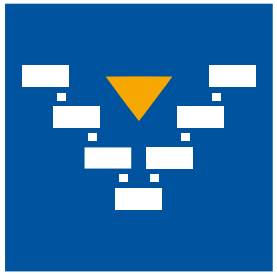


Introduction to

Embedded Systems

Prof. Dr.-Ing. Stefan Kowalewski | Julius Kahle, M. Sc.

Summer Semester 2025

Exercise 4 – Real Time

Overview

- **Basics**
- **OSEK**
- **Real Time and Resources**
- **Periodic Scheduling**

Basics

Task 1:

Task 1:

a) Name the two requirements needed for real time.

① computation finishes before Deadline

② computation is correct

for embedded systems

① reaction/response time must be ≤ maximal value

real time = in time

Task 1:

DL = Deadline

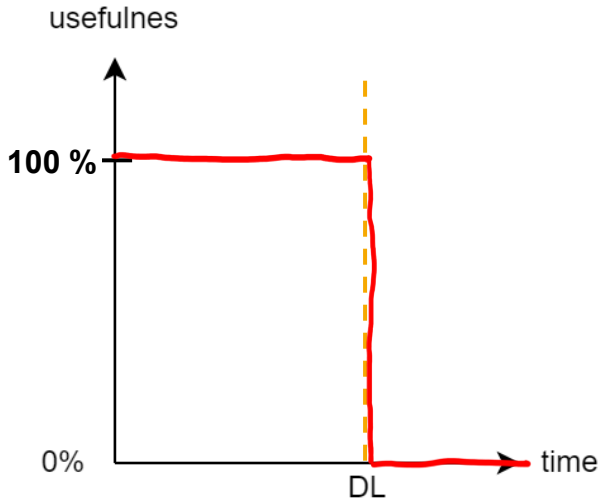
b) Explain the terms:

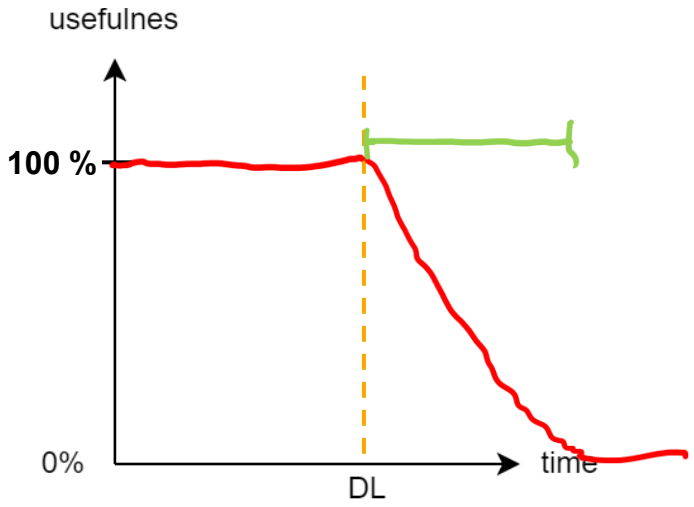
Hard real time

- useless after DL
  - airbag
- more important to adhere DL

Soft real time

- still useful for certain amount of time
  - marker on navi
- less important to adhere DL



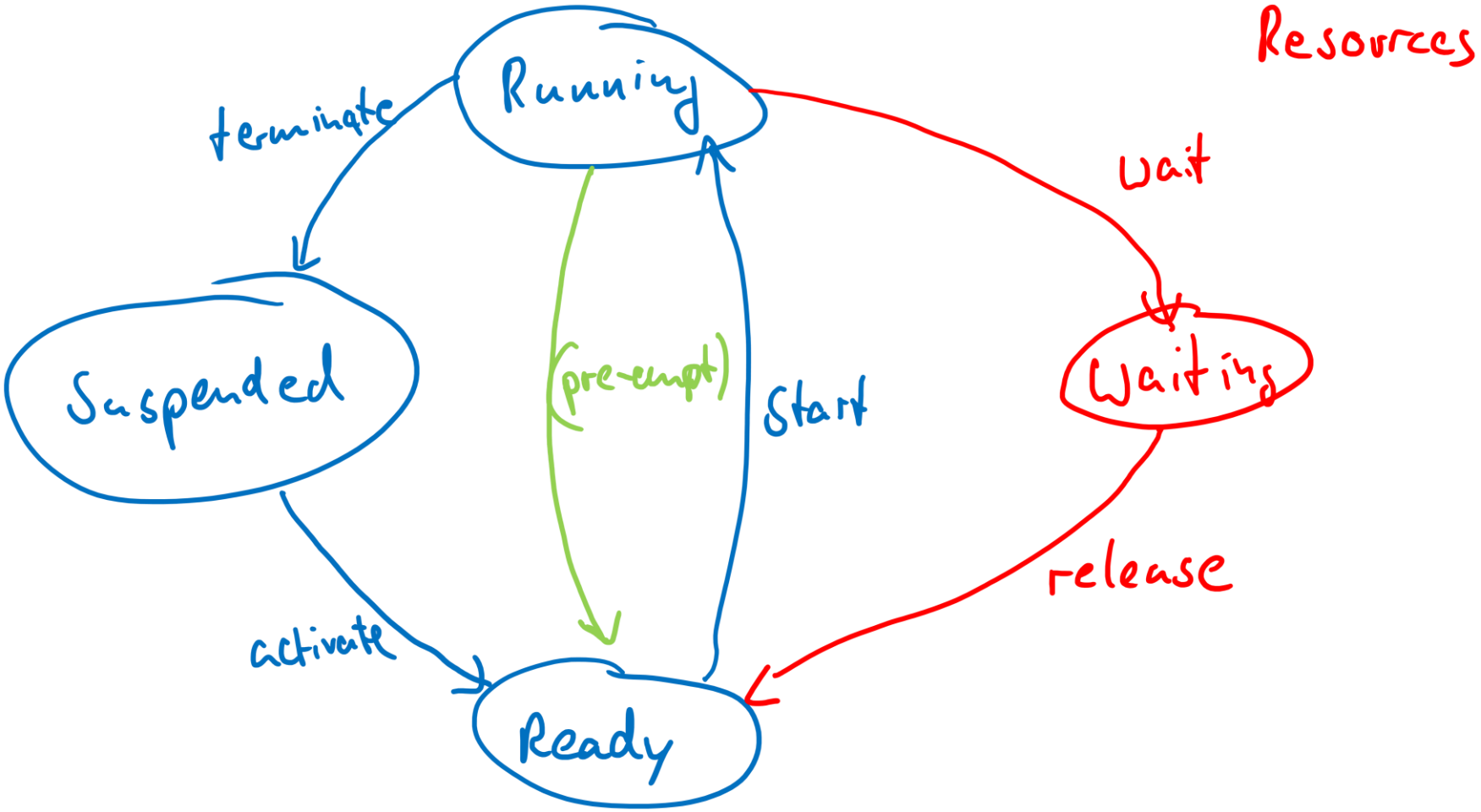


OSEK

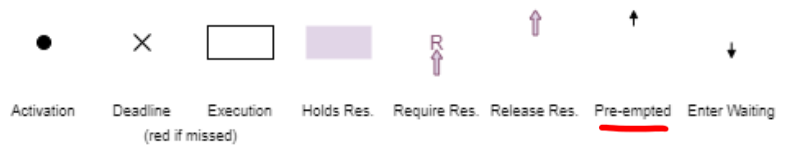
Task 2

Task 2

a) Sketch the extended OSEK task model.

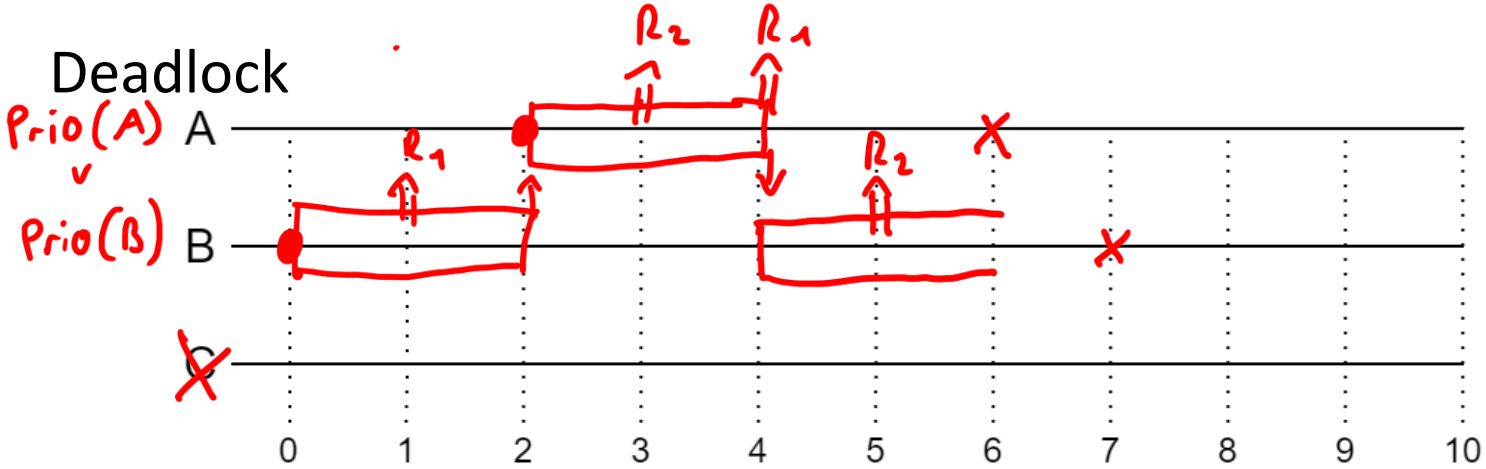


Task 2



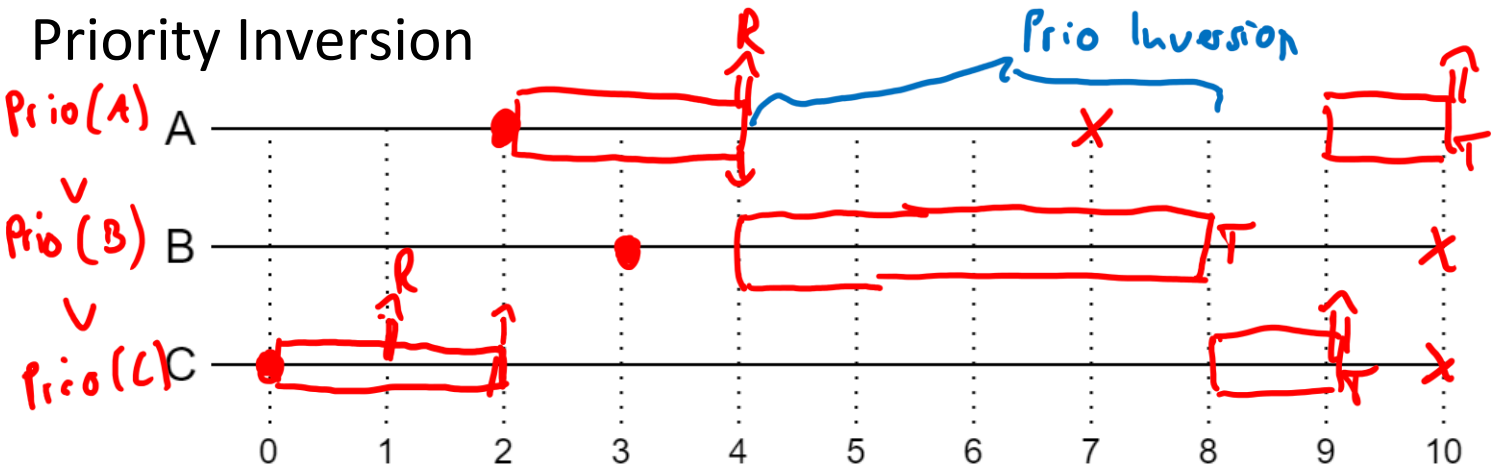
b) How many processes and resources are needed for:

Deadlock



min 2 Proc
min 2 Res

Priority Inversion

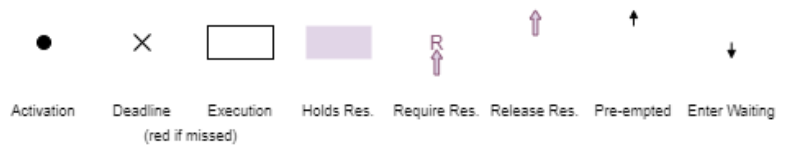


min 3 Proc
min 1 Res

Real Time and Resources

Task 3

Task 3

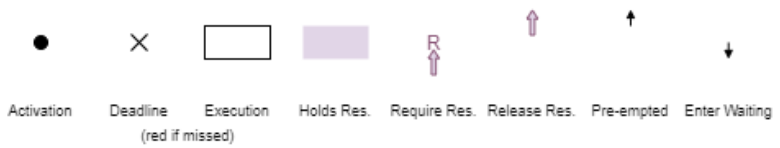


Schedule these tasks (sorted by priority; first task has highest priority)

Task	Execution Pattern	Deadline (absolute)
Prio(1) Task A	Start @ 5 | runs 1 | Req. | runs 1 | T	DL @ 10
Task B	Start @ 3 | runs 1 | T	DL @ 5
Task C	Start @ 5 | runs 3 | T	DL @ 13
Prio(0) Task D	Start @ 1 | runs 3 | Req. | runs 3 | T	DL @ 13

Activate

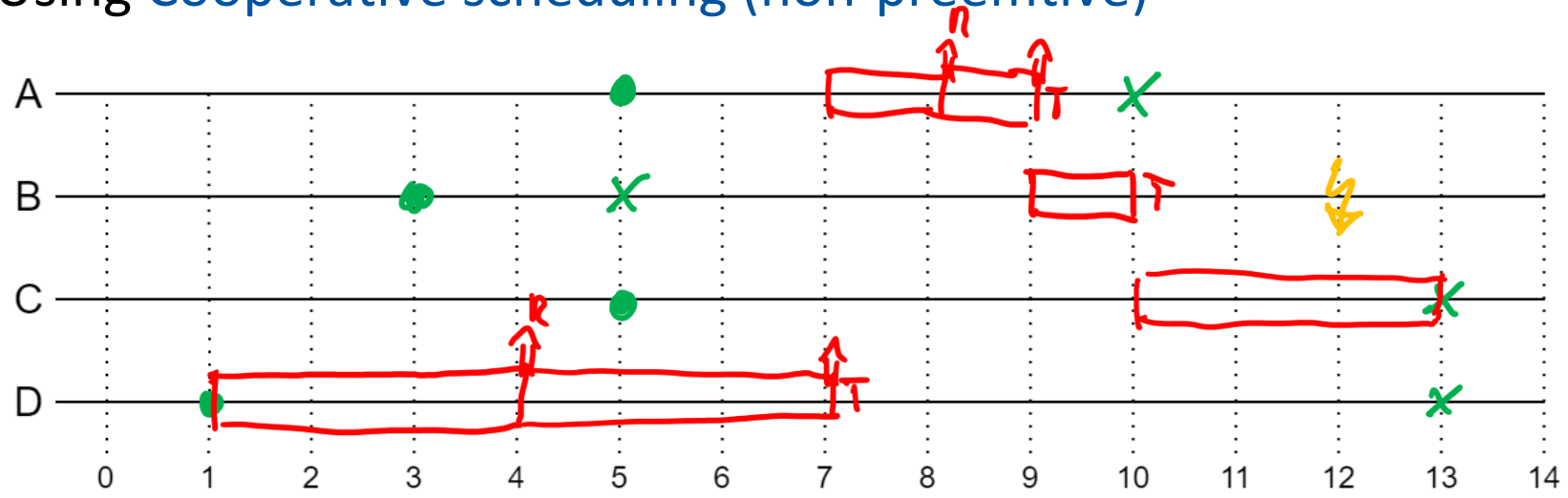
Task 3



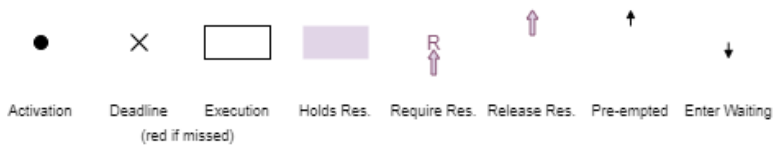
Task	Execution	Deadline (absolute)
Task A	Start @ 5 | runs 1 | Req. | runs 1 | T	DL @ 10
Task B	Start @ 3 | runs 1 | T	DL @ 5
Task C	Start @ 5 | runs 3 | T	DL @ 13
Task D	Start @ 1 | runs 3 | Req. | runs 3| T	DL @ 13

Activate

Using Cooperative scheduling (non-preemtive)



Task 3

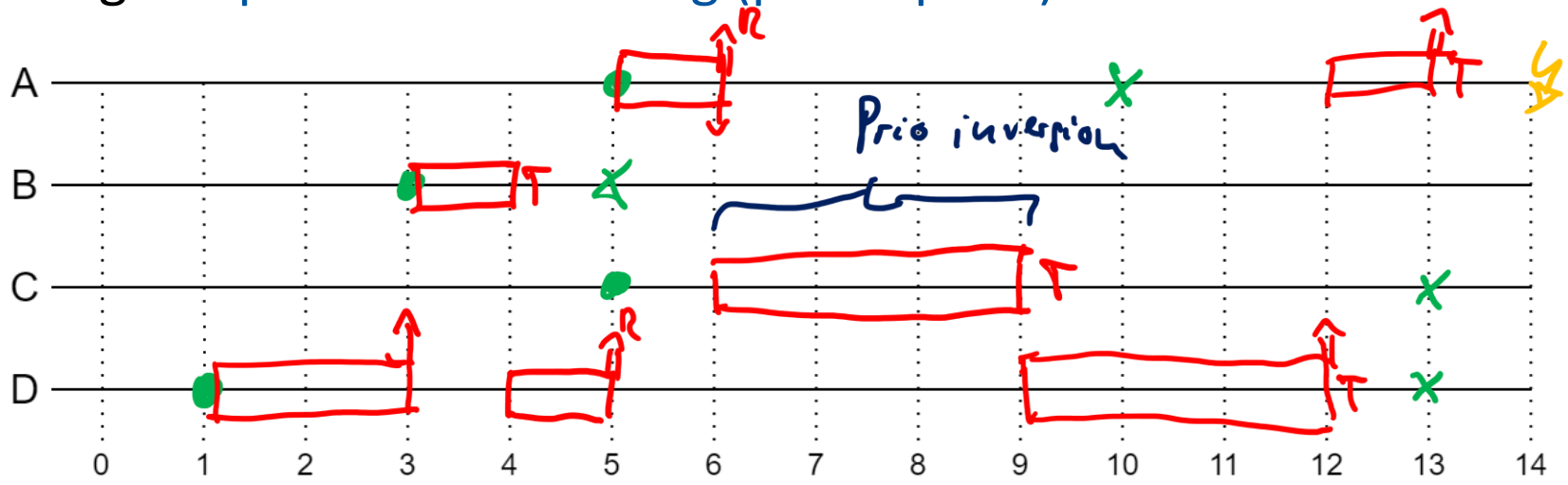


Task	Execution	Deadline (absolute)
Task A	Start @ 5 | runs 1 | Req. | runs 1 | T	DL @ 10
Task B	Start @ 3 | runs 1 | T	DL @ 5
Task C	Start @ 5 | runs 3 | T	DL @ 13
Task D	Start @ 1 | runs 3 | Req. | runs 3| T	DL @ 13

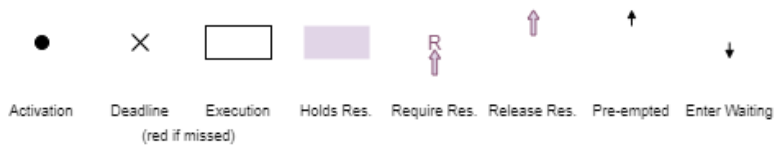
Act.

preemptive

Using ~~Cooperative~~ scheduling (~~preemptive~~)

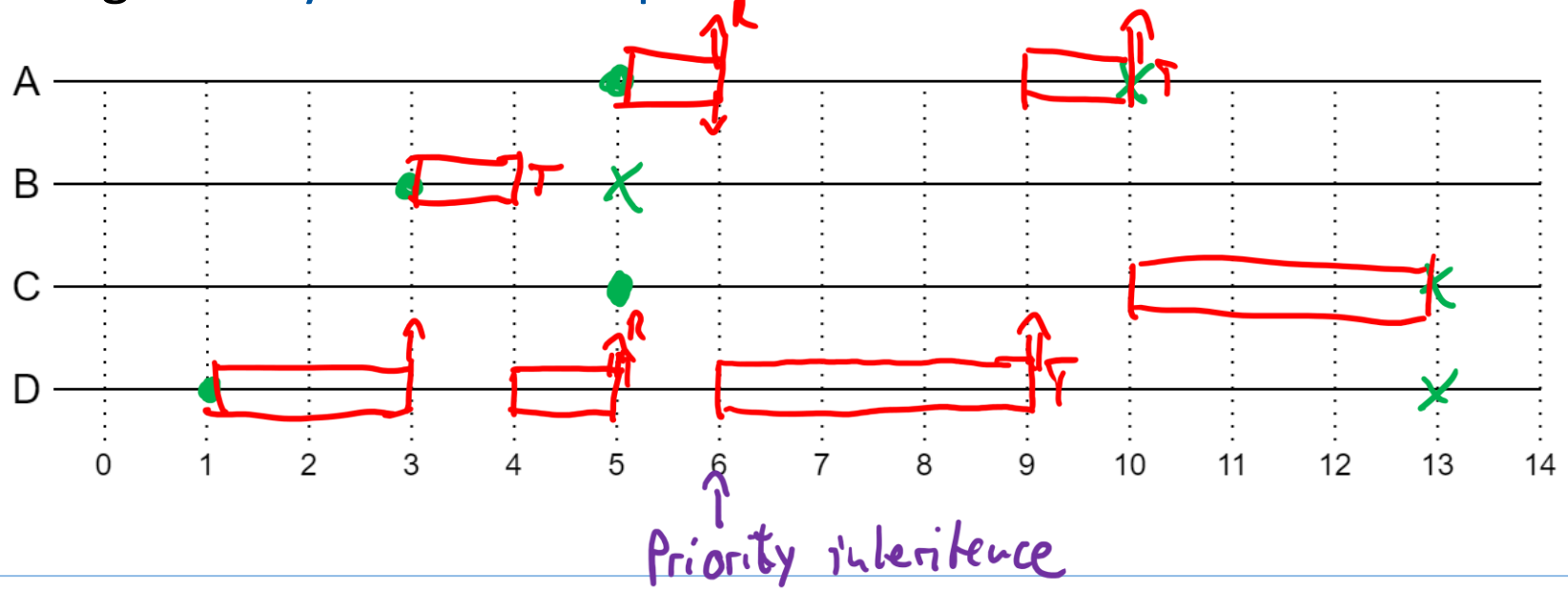


Task 3

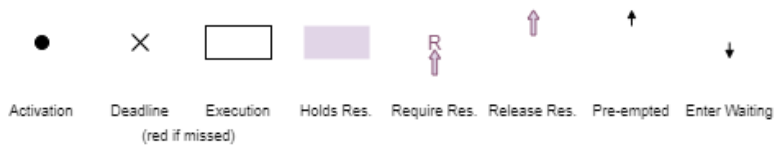


Task	Execution	Deadline (absolute)
Task A	Start @ 5 | runs 1 | Req. | runs 1 | T	DL @ 10
Task B	Start @ 3 | runs 1 | T	DL @ 5
Task C	Start @ 5 | runs 3 | T	DL @ 13
Task D	Start @ 1 | runs 3 | Req. | runs 3| T	DL @ 13

Using Priority inheritence protocol

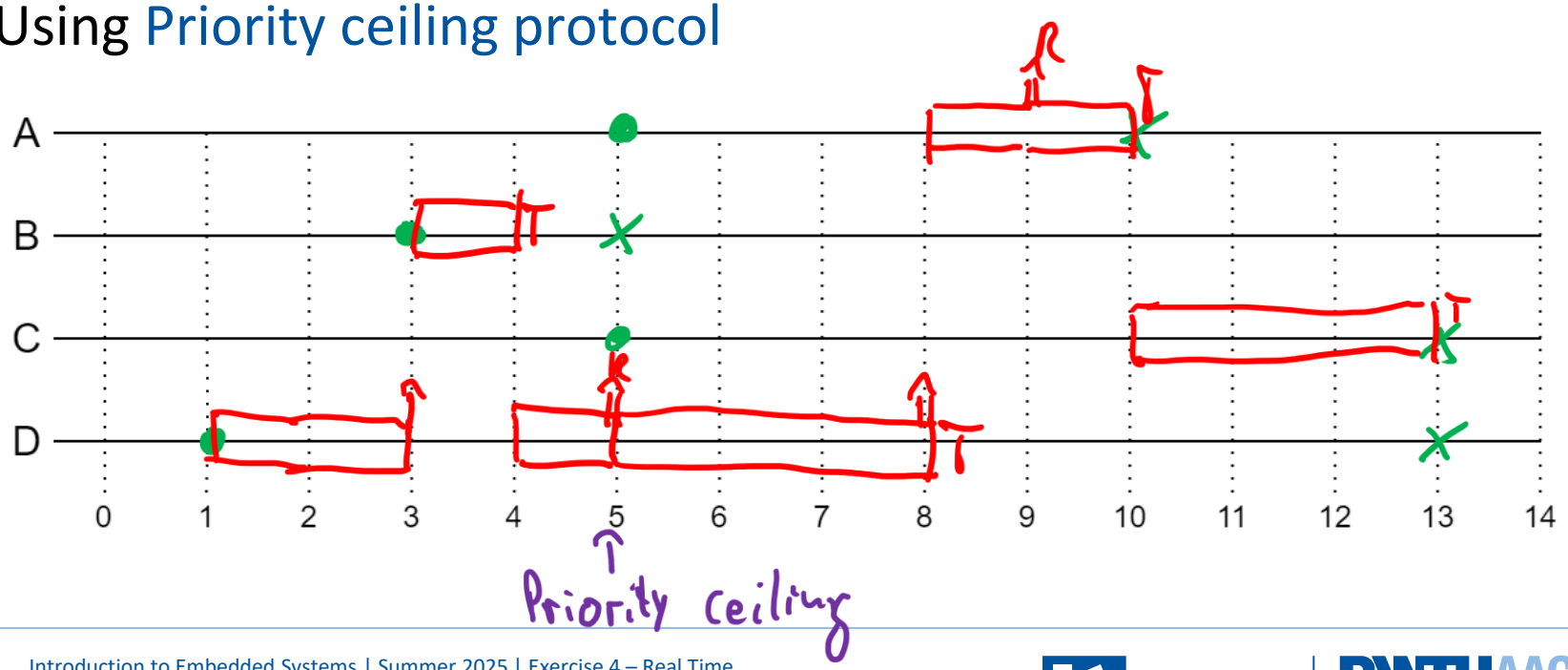


Task 3



Task	Execution	Deadline (absolute)
Task A	Start @ 5 | runs 1 | Req. | runs 1 | T	DL @ 10
Task B	Start @ 3 | runs 1 | T	DL @ 5
Task C	Start @ 5 | runs 3 | T	DL @ 13
Task D	Start @ 1 | runs 3 | Req. | runs 3| T	DL @ 13

Using Priority ceiling protocol



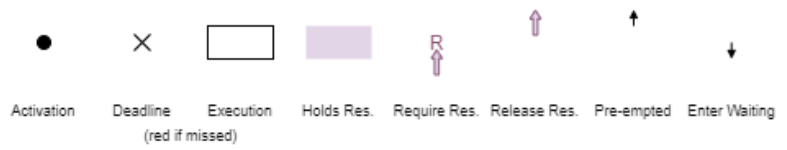
Periodic Scheduling

Task 4

Task 4

$P_i := (\tilde{C}_i, T_i, DL_i)$

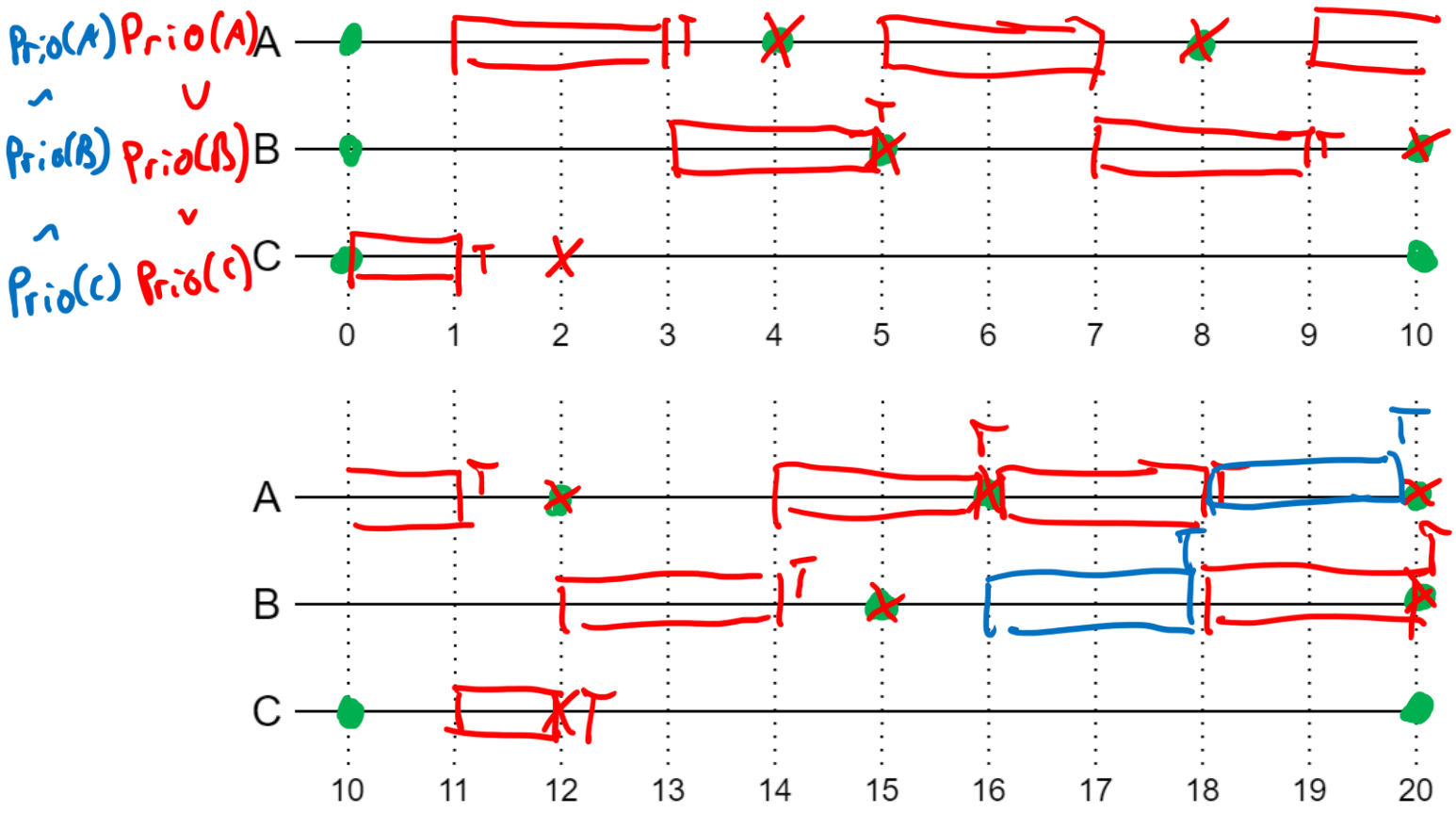
EDF



a) Use earliest deadline first to schedule this task system.

A := (4, 2, 4), B:= (5, 2, 5), C:= (10, 1, 2)

all tasks activated @0



RMS

Rate Monotonic Scheduling

$Rate = \frac{1}{\tilde{C}_i}$

Task 4

b) Why is the following task system not schedulable?

A:= (3, 2, 2), B:= (6, 2, 7), C:= (10, 3, 10)

C_i T_i

$$\text{Utilization } (U) = \sum_{i=1}^{n} \frac{T_i}{T_i}$$

$$= \frac{2}{3} + \frac{2}{6} + \frac{3}{10} = 1{,}3 \not\leq 1$$

$\frac{2}{3} + \frac{2}{6}$: 1 — $\frac{3}{10}$: 0,3

not schedulable

necessary condition for schedulability

⇒ not sufficient!