Real-Time Systems

Exam Preparation

Abstract

The following exercises are a compilation from the exams Real-Time Systems summer semester 2017 and 2020. In addition, they contain exercises that were given by the lecturer as hints for possible examination tasks, but were not part of the exams in this form. Exercises from the 2017 and 2020 exams can be identified by the points to be achieved for them. The correctness of all exercises is not guaranteed and it is recommended to check everything again by yourself.

Exercise 1: (11 Points)

a) Name the two most important requirements for real-time systems. (2P)

1.

2.

b) Name the three types of hardness of real-time systems and make a short explanation. Give an example of each. (3P)

1.

2.

3.

c) Specify the remaining criteria (not those of b) to classify real-time systems. (3P)

1.

2.

3.

4.

5.

6.

7.

c) Name and explain all possible tasks of a real-time system.

1.

2.

3.

d) Explain and draw the idealized model of real-time systems.

e) Explain how a timer works.

f) Explain how a stopwatch works.

g) Explain how a stopwatch with laps works.

h) Name additional requirements of a real-time system

i) Requirements for real-time-clocks

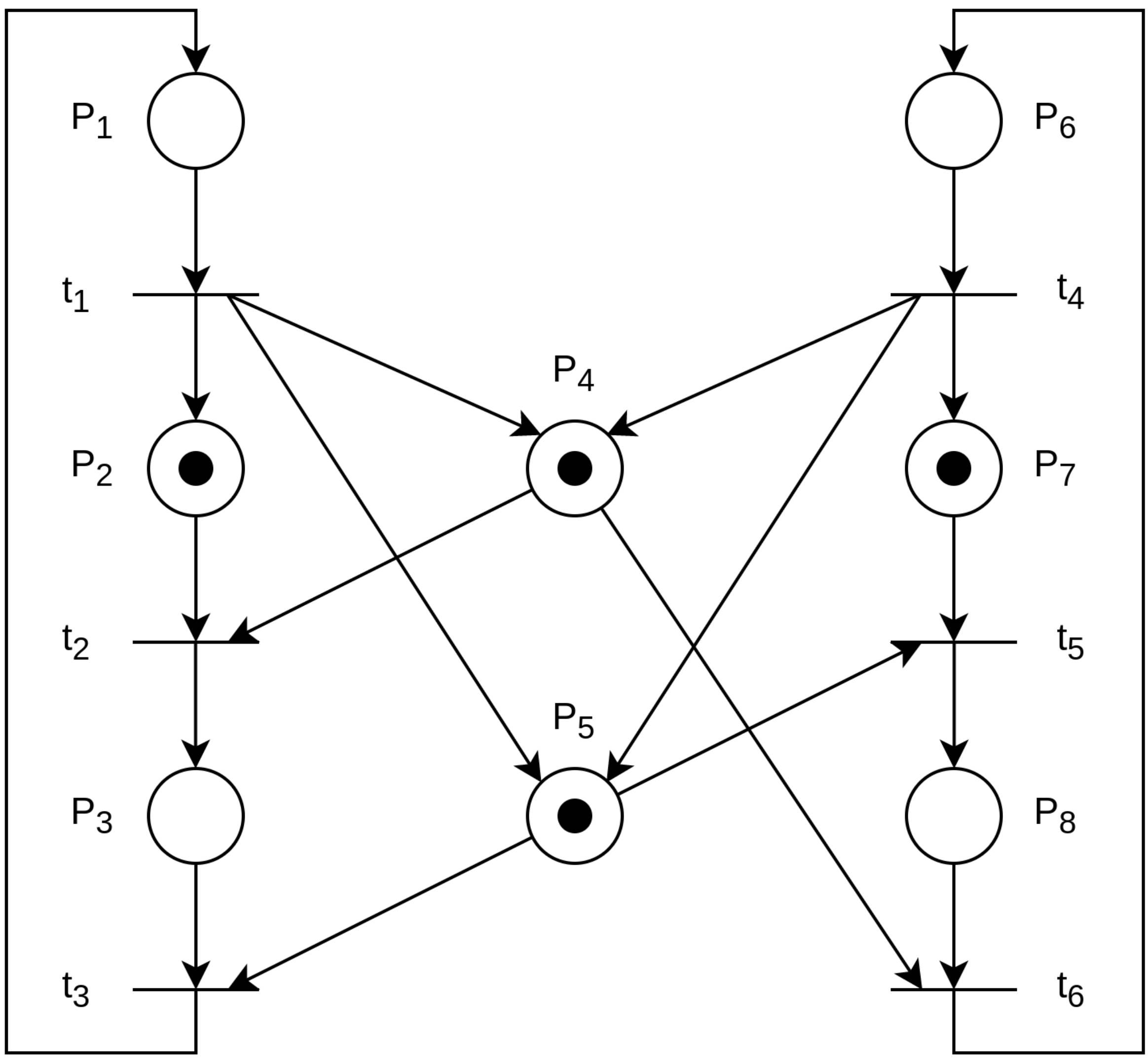
j) Name an advantage and possible issues of concurrency

Exercise 2: Earliest Deadline First (Take your time and work carefully)

Schedule taskset $\mathrm{T}^{\mathrm{k}} = (\Delta \mathrm{t_{exec}}^{\mathrm{k}}, \Delta \mathrm{t_{dead}}^{\mathrm{k}})$: $\mathrm{T}^1 = (1, 3)$; $\mathrm{T}^2 = (2, 7)$; $\mathrm{T}^3 = (3, 11)$ for slice 0..26

Exercise 3: (10 Points)

Please analyse the Petri net on the next page:



Please prepare a <u>detailed</u> reachability graph with all places! (4P)
(You may use the back side of the paper, if necessary)

Please verify or falsify the following properties!
<u>Underline the result</u> and give a short argument for your decision.

The net is alive/ is not alive (0,66P)

because (1,33P):

The net is safe/ is not safe (0,66P)

because (1,33P):

All places are reachable/ some places are not reachable (0,66P)

because (1,33P):

Exercise 4: (8 Points) Petri-Net for traffic lights (do not care about times of the traffic lights)

(Hint: No inhibitors-net, no timed Petri-net, no colored Petri-net)

a) Please model a Petri-Net that represents the different statuses of traffic lights for pedestrians. It has two states <red> and <green> (2P)

b) Please model a Petri-Net that represents the different statuses of traffic lights for automotive traffic.
It has three states: <red>, <yellow> and <green> (3P) The switching sequence is <red> to <yellow> to <green> to <yellow> to <red>. There is one color only at a time.

c) Combine the two Petri-Nets of a) and b) in that way that pedestrians have <green> only when the automotive traffic has <red> (make them mutual exclusive). (3P)

Exercise 5: Bicycle (3 Points)

The rate of the foot pedal driven by the driver is between 0 and 120 tunes per minute.

The gear ("Gangschaltung") is 52 teeth (front) to 13 teeth (back) = 4:1. That means the wheels of the bicycle turn 4 times faster than the foot pedal.

The rate of heart-beat of the driver is between 60 and 180 beats per minute.

Your task is to design a bicycle computer that measures the rates of the:

1. foot pedal, 2. back wheel, 3. the heart beat of the driver.

Please calculate the minimum period (in seconds) for the three parameters:

$\min(\Delta t_{\text{per}}{}^{\text{foodpedal}}) \quad =$

$\min(\Delta t_{\text{per}}{}^{\text{wheel}}) \quad =$

$\min(\Delta t_{\text{per}}{}^{\text{heartbeat}}) \quad =$

Exercise 6a: (in relation to No 5. You can solve 6 without 5)

If you use units of 1/24 seconds you will get the following paramters.

$\min(\Delta t_{per}{}^{\text{foodpedal}}) \quad = 12$

$\min(\Delta t_{per}{}^{\text{wheel}}) \quad = 3$

$\min(\Delta t_{per}{}^{\text{heartbeat}}) \quad = 8$

The execution times for the computer are given:

$\min(\Delta t_{exec}{}^{\text{foodpedal}}) \quad = 2$

$\min(\Delta t_{exec}{}^{\text{wheel}}) \quad = 1$

$\min(\Delta t_{exec}{}^{\text{heartbeat}}) \quad = 3$

This leads to $T^i = (\Delta t_{exec}{}^i, \Delta p^i)$

$T^{\text{footpedal}} = (2, 12)$; $\quad T^{\text{wheel}} = (1, 3) \quad T^{\text{heartbeat}} = (3, 8)$

Please calculate the load of the CPU for all three tasks and in total (2P)

(you need to write down the complete calculation, and not the result only)

What does this mean for the schedulability (Yes/No/Perhaps) any why) (2P)

Exercise 6b: refering to Exercise 6a

Please assign priorities to the three tasks (a high number should be a high priority) (2P)

$$\begin{aligned} \min(\text{proprity}^{\text{footpedal}}) &= \\ \min(\text{proprity}^{\text{wheel}}) &= \\ \min(\text{proprity}^{\text{heartbeat}}) &= \end{aligned}$$

Schedule the problem on the solution sheet with RMS, interruptible tasks.

Fill in A or B or C or nothing in row CPU for slices 0..23 (6P)

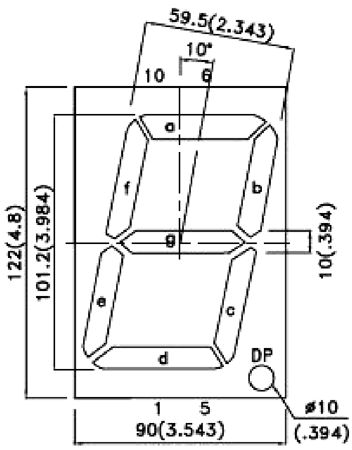
Exercise 7: (4 Points)

You shoud use a microcontroller (e.g STR9) and a 7-segment-display. Given the specification of a 7-segment-display connected to a microcontroller-ports.

The seven segment display is connected to the following ports:

- segment g at GPIO 4.0
- segment f at GPIO 4.1
- segment e at GPIO 4.2
- segment d at GPIO 4.3
- segment c at GPIO 4.4
- segment b at GPIO 4.5
- segment a at GPIO 4.6

The segments of this board are active low.
You can set all bits at one time by: `GPIO4 -> DATA[255<<2] = BITARRAY`, e.g to display a 1: `1100 1111 = 0xCF -> GPIO4 -> DATA[255<<2] = 0xCF`



What hexadecimal number you need to store to display a 4.

`GPIO4 -> DATA[255<<2] = 0x_____.`

What hexadecimal number you need to store to display a 2.

`GPIO4 -> DATA[255<<2] = 0x_____.`

Exercise 8a: (4 Points)

Name at least 8 different functional units that are integrated on a microcontroller development chip.

Exercise 8b: (extra 2 Points)

What is a minor page fault? Explain it one or two senteces (1,5P)

Why are minor page faults unwanted in real time systems and are prevented by `mlockall(MCL_CURRENT | MCL FUTURE)`? (0,5P)

Exercise 9: Explain what active high and active low is.

Exercise 10: Explain what the Waiting Time Δt_{wait} is.

Exercise 11: Explain what the Execution Time Δt_{exec} is.

Exercise 12: Explain what the Response Time Δt_{resp} is.

Exercise 13: Explain what the Slack Time Δt_{slack} is.

Exercise 14: Explain what the Tardiness Δt_{tard} is.

Exercise 15: Explain where information about every process in the computer is stored.

Exercise 16: When is the Process Control Block created and when it is removed?

Exercise 17: Where is the Process Control Block stored?

Exercise 18: What information does a Process Control Block store?

Exercise 19: Explain what an interrupt is and how it works.

Exercise 20: Name and explain the types of interrupts.

Exercise 21: Explain what a quantization error is.

Exercise 22: Explain what the sample theorem is.

Exercise 23: Explain what anti-aliasing is.