TECHNISCHE UNIVERSITEIT DELFT
Faculteit Elektrotechniek, Wiskunde en Informatica

IN4343 – Real Time Systems

April 13th 2017, from 09:00 to 12:00

Koen Langendoen

Question:	1	2	3	4	5	6	Total
Points:	15	25	5	15	10	25	95
Score:							

- This is a closed book exam
- You may use a **simple** calculator only (i.e. graphical calculators are not permitted)
- Write your answers with a black or blue pen, not with a pencil
- **Always justify your answers, unless stated otherwise**

- The exam covers the following material:
  (a) chapters 1-6, 8-9 of the book "Hard Real-Time Computing Systems (3rd ed)" by G. Buttazzo
  (b) the paper "The Worst-Case Execution-Time Problem" by Wilhelm et al. (except Section 6)
  (c) the paper "Transforming Execution-Time Boundable Code into Temporally Predictable Code" by P. Puschner
  (d) the paper "Best-case response times and jitter analysis of real-time tasks" by R.J. Bril, E.F.M. Steffens, and W.F.J. Verhaegh

IN4343 Real-Time Systems *Cheat Sheet*

Liu and Layland (LL) bound		$U_{lub}^{RM} = n(2^{1/n} - 1)$
Hyperbolic (HB) bound		$\prod_{i=1}^{n}(U_i + 1) \leq 2$
Response Time Analysis		$WR_i = C_i + \sum_{k=1}^{i-1} \left\lceil \frac{WR_i + AJ_k}{T_k} \right\rceil C_k$ $BR_i = C_i + \sum_{k=1}^{i-1} \left(\left\lceil \frac{BR_i - AJ_k}{T_k} \right\rceil - 1 \right)^{+} C_k$ $w^{+} = \max(w, 0)$
Processor Demand		$g(t_1, t_2) = \sum_{r_i \geq t_1}^{d_i \leq t_2} C_i \qquad g(0, L) = \sum_{i=1}^{n} \left\lfloor \frac{L + T_i - D_i}{T_i} \right\rfloor C_i$
	schedulability	$\forall L \in D, \quad g(0, L) \leq L$ $D = \{d_k \mid d_k \leq \min(H, \max(D_{max}, L^*))\}$ $H = lcm(T_1, \ldots, T_n)$ $L^* = \frac{\sum_{i=1}^{n}(T_i - D_i)U_i}{1 - U}$
Polling Server	schedulability	$U_{lub}^{RM+PS} = U_s + n \left[\left(\frac{2}{U_s + 1} \right)^{1/n} - 1 \right]$ $\prod_{i=1}^{n}(U_i + 1) \leq \frac{2}{U_s + 1}$
	response time	$R_a = C_a + \Delta_a + F_a(T_s - C_s)$ $\Delta_a = \left\lceil \frac{r_a}{T_s} \right\rceil T_s - r_a \qquad F_a = \left\lceil \frac{C_a}{C_s} \right\rceil - 1$
Deferrable Server	schedulability	$U_{lub}^{RM+DS} = U_s + n \left[\left(\frac{U_s + 2}{2U_s + 1} \right)^{1/n} - 1 \right]$ $\prod_{i=1}^{n}(U_i + 1) \leq \frac{U_s + 2}{2U_s + 1}$
	response time	$R_a = C_a + \Delta_a - C_0 + F_a(T_s - C_s)$ $C_0 = \min(C_s(r_a), \Delta_a)$ $\Delta_a = \left\lceil \frac{r_a}{T_s} \right\rceil T_s - r_a \qquad F_a = \left\lceil \frac{C_a - C_0}{C_s} \right\rceil - 1$
NP scheduling	level-i busy period	$L_i = B_i + \sum_{h=1}^{i} \left\lceil \frac{L_i}{T_h} \right\rceil C_h \qquad N_i = \left\lceil \frac{L_i}{T_i} \right\rceil$ $B_i = \max_{j>i}\{C_j\}$
	response time	$s_{ik} = B_i + (k-1)C_i + \sum_{h=1}^{i-1} \left(\left\lfloor \frac{s_{ik}}{T_h} \right\rfloor + 1 \right) C_h$ $R_{ik} = (s_{ik} + C_i) - (k-1)T_i$ $R_i = \max_{k \in [1, N_i]} \{R_{ik}\}$
Elastic Model	utilization	$\forall i \quad U_i = U_{i0} - (U_0 - U_d)\frac{E_i}{E_S} \qquad$ where $E_S = \sum_{i=1}^{n} E_i$

Question 1 [15 points]

Given the following set of aperiodic, preemptable tasks

	A	B	C	D	E	F	G
r_i	0	0	5	0	5	5	5
C_i	3	1	1	2	4	1	2
d_i	10	10	15	10	15	15	15

with precedent constraints:

$$A\rightarrow C,\ B\rightarrow C,\ B\rightarrow F,\ C\rightarrow E,\ C\rightarrow F,\ D\rightarrow F,\ E\rightarrow G.$$

To determine if there is a feasible schedule EDF* can be employed, which transforms the precedence constraints into timing constraints by updating the release times and deadlines of the tasks.

(a) 2 points Which criterion does EDF* optimize?

Solution: The maximum lateness, L_{max}

(b) 3 points Explain that although EDF considers deadlines only for determining the priority among tasks, the release times must be adjusted too.

Solution: Consider a precedence X→Y. If task Y is released before task X and there are no other ready tasks with a deadline earlier than Y, then EDF will run Y (while it should wait until X finishes). That can be fixed by ensuring that the release time of Y is larger than that of X.

(c) 3 points Transform the release times according to the precedence constraints.

Solution: Apply the max rule starting at the roots.

	A	B	C	D	E	F	G
r_i	0	0	5	0	6	6	10

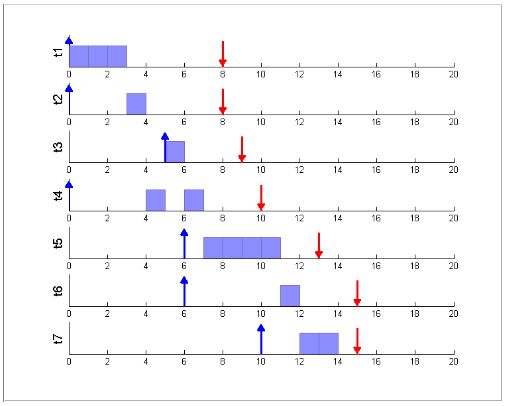
(d) 3 points Transform the deadlines according to the precedence constraints.

Solution: Apply the min rule starting at the leaves.

	A	B	C	D	E	F	G
d_i	8	8	9	10	13	15	15

(e) 4 points Determine if the task set is feasible. If so, what is the minimal slack? If not, what is the maximum lateness?

Solution: The schedule is feasible, and has a minimal slack of 1 (Lmax = -1) for task G (or F). Note that the picture shows the transformed release times and deadlines, but the task lateness must be computed using the original deadlines.



Question 2 [25 points]

Consider the following task set:

	C_i	D_i	T_i
τ_1	2	3	5
τ_2	4	6	8
τ_3	4	32	40

(a) 10 points Use the processor demand criterion to demonstrate that the above task set is unfeasible under EDF scheduling.

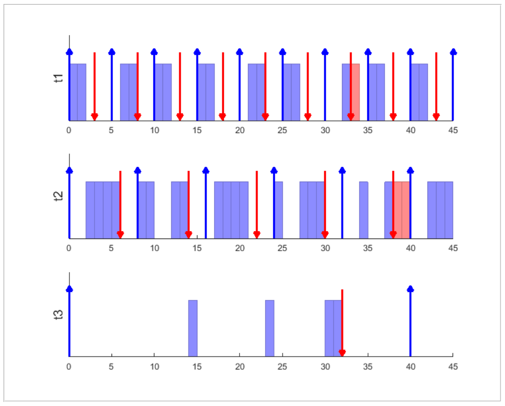
Solution: Processor utilization is 1 (!), so L^* can't be computed and we need to go as far as the Hyper period (40).

	demand
g(0,3)	2
g(0,6)	6
g(0,8)	8
g(0,13)	10
g(0,14)	14
g(0,18)	16
g(0,22)	20
g(0,23)	22
g(0,28)	24
g(0,30)	28
g(0,32)	32
g(0,33)	34

Task τ_1 (which must complete by $t = 33$) misses its deadline!

(b) 5 points Determine the worst-case response times of the three tasks. *Hint: draw a time line.*

Solution:



Task	Response time
τ_1	4
τ_2	8
τ_3	32

Alternative solution is to avoid a context switch at $t = 35$ allowing task τ_2 to finish within its deadline, and task τ_1 to incur a second deadline violation.

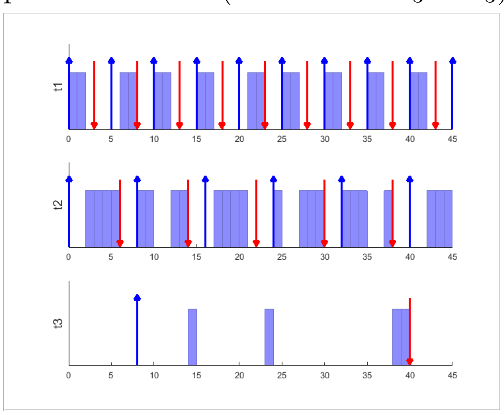
Task	Response time
τ_1	5
τ_2	6
τ_3	32

(c) 2 points Releasing all the tasks at once is creating the highest demand at the start of the schedule (à la the critical instant for RM scheduling). Articulate why starting task τ_3 later could be beneficial.

Solution: The "slack" between the deadline and the next period of task τ_3 can only be used at the end of the Hyper period, while it may be of better use at the beginning.

(d) 5 points Can you find a phase offset for task τ_3 that makes the schedule feasible under EDF?

Solution: Yes we can! The utilization is 1, so there must be enough space for hosting task τ_3 from the first bit of execution (at $t = 14$) until the end of the Hyper period (at $t = 40$). As task τ_3's relative deadline is larger than the gap ($32 > 26 = 40 - 14$) any phase between 8 ($= 40 - 32 = T_3 - D_3$) and 14 will do:



(e) 3 points Provide a revised definition of the processor demand $g(0, L)$ that takes phase offsets (for all tasks) into account.

Solution:

$$\boxed{g(0,L) = \sum_{i=1}^{n} \left\lfloor \left(\frac{L - \phi_i + T_i - D_i}{T_i} \right)^+ \right\rfloor C_i}$$

Question 3 [5 points]

When jitter is not a factor, the best case response time of tasks under RM scheduling can be determined by iteratively computing the least fixed point of the following recursive equation:

$$BR_i = C_i + \sum_{k=1}^{i-1} \left(\left\lceil \frac{BR_i}{T_k} \right\rceil - 1 \right) C_k$$

(a) $\boxed{\text{3 points}}$ Explain why one cannot simply round down, as in $\left\lfloor \frac{BR_i}{T_k} \right\rfloor$, but has to round up and subtract 1.

Solution: The expressions $\lfloor x \rfloor$ and $\lceil x \rceil - 1$ and differ only for integral numbers of x. That is if BR_i is a multiple of T_k when constructing the optimal instant for task i. In that case task k will be executed first as it has higher priority, so there is no point in starting task i at that moment; one should wait until k finishes i.e., start task i later so k can interfere one time less. In other words, when rounding down the iterative procedure would stop too early when hitting such an integral point.

(b) $\boxed{\text{2 points}}$ Show that $BR_i^* = C_i + \sum_{k=1}^{i-1} \frac{BR_i^*}{T_k} C_k$ is a valid starting point.

Solution: We need to show that BR_i^* is larger than the true solution. It holds that $\lceil x \rceil - 1 < x \le \lceil x \rceil$, also for x being an integral number, from which it follows that

$$BR_i^* = C_i + \sum_{k=1}^{i-1} \left(\frac{BR_i^*}{T_k} \right) C_k > C_i + \sum_{k=1}^{i-1} \left(\left\lceil \frac{BR_i^*}{T_k} \right\rceil - 1 \right) C_k = BR_i$$

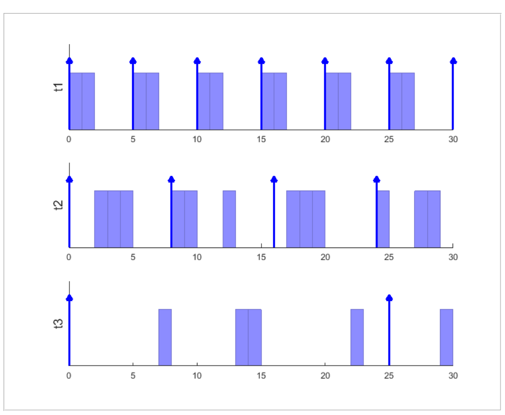
Question 4 [15 points]

Consider jitter analysis (RM scheduling) for the following task set:

	C_i	T_i
τ_1	2	5
τ_2	3	8
τ_3	4	25

(a) $\boxed{\text{3 points}}$ Draw the critical instant for task τ_3 and report its worst-case response time.

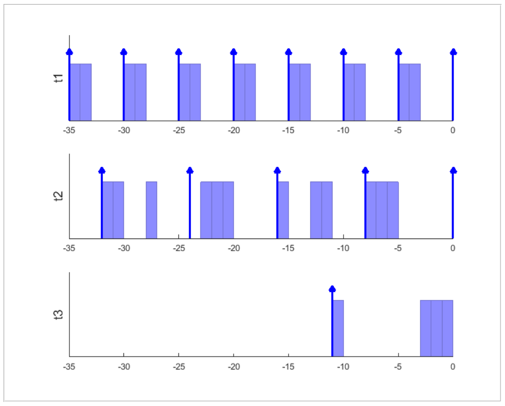
Solution:



Task	Worst case
τ_1	2
τ_2	5
τ_3	23

(b) $\boxed{\text{4 points}}$ Draw the optimal instant for task τ_3 and report its best-case response time.

Solution:



Task	Best case
τ_1	2
τ_2	3
τ_3	11

(c) $\boxed{\text{3 points}}$ Derive the response jitter of **all** three tasks.

Solution:

Task	Response jitter
τ_1	0
τ_2	2
τ_3	12

(d) $\boxed{\text{5 points}}$ If task τ_2 would be subject to an activation jitter of one (1), what changes to the response jitter of the three tasks? *Hint: a time line tells more than a thousand equations.*

Solution: For task τ_1 (highest prio) and τ_2 (activation jitter is **not** part of the response time) nothing changes. Then for task τ_3, the worst-case response time does not change as there is an extra unit of slack at $t = 23$ that can accommodate the 1 unit of jitter by task τ_2 (whose deadline moves from 24 to 23). Likewise the best case does not change either as the extra unit of space does release task τ_2 earlier at $t = -9$, but it cannot run until $t = -8$ as before (task τ_1 has precedence). Thus, response jitter stays at 12!

Question 5 [10 points]

When mixing periodic and aperiodic tasks one can make use of a priority server to schedule the aperiodic tasks. Consider the following periodic tasks and aperiodic jobs (under EDF):

	C_i	D_i	T_i
τ_1	2	4	8
τ_2	3	6	10
τ_3	3	9	12

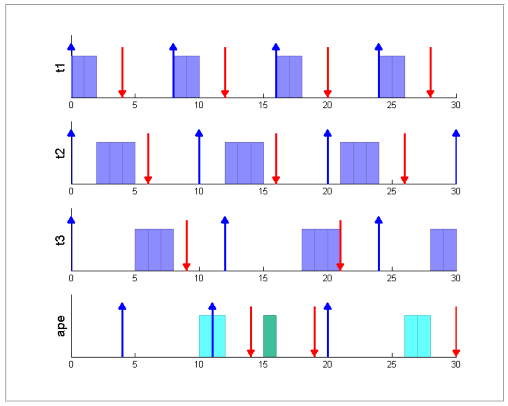
	a_i	C_i
J_1	4	2
J_2	11	1
J_3	20	2

(a) $\boxed{\text{5 points}}$ Determine the response times for the three jobs when being served by (plain) TBS.

Solution:

Max server utilization is $U_S = 1 - U_p = \frac{1}{5}$.

	d_i	f_i	R_i
J_1	14	12	8
J_2	19	16	5
J_2	30	28	8

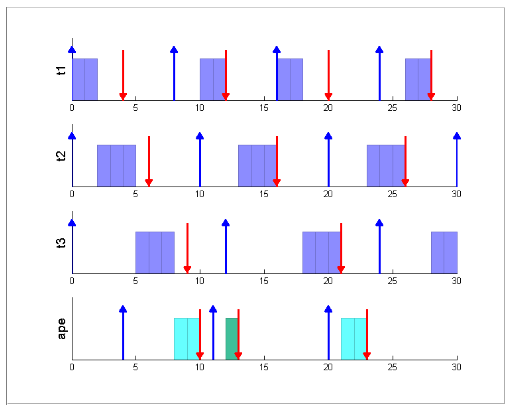


(b) 5 points Determine the response times for the three jobs when being served by **optimized** TBS.

Solution:

Iterate with $d_{i+1} = f_i$ until convergence

	f_1	f_2	f_3	R_i
J_1	12	10		6
J_2	16	13		2
J_3	28	26	23	3



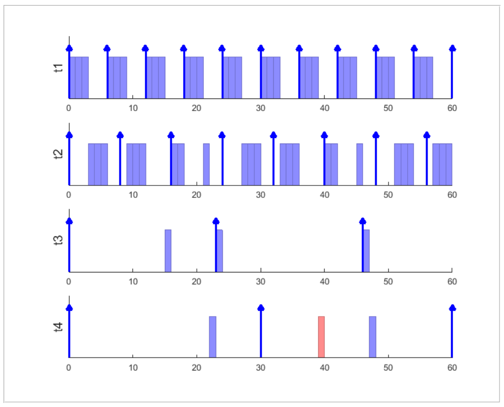
Question 6 [25 points]

Consider the following set of periodic tasks:

	C_i	T_i
τ_1	3	6
τ_2	3	8
τ_3	1	23
τ_4	2	30

(a) $\boxed{\text{7 points}}$ What is the maximum lateness of this task set under RM scheduling?

Solution: $L_{max} = 10$, so unfeasible



When switching to **non-preemptive** RM scheduling the self-pushing phenomenon complicates matters, and may require the analysis to extend beyond the first period of a task.

(b) $\boxed{\text{8 points}}$ Report how many periods we need to consider for the above tasks.

Solution: We can't take a shortcut as the task set is not feasible under preemptive scheduling, so we have to compute the level-i busy times.

Task	Blocking	Level-i busy	Number
τ_1	3	-	1
τ_2	2	23	3
τ_3	2	40	2
τ_4	0	88	3

(c) $\boxed{\text{10 points}}$ Report the worst-case response times for the four tasks, and conclude if the task set is feasible.

Solution:

Task	Worst case
τ_1	6
τ_2	8
τ_3	24
τ_4	24

Task τ_3 misses its deadline, so unfeasible.