jian-jia.chen [☺] tu-dortmund.de
georg.von-der-brueggen [☺] tu-dortmund.de
wen-hung.huang [☺] tu-dortmund.de

Exercises for Lecture
Real-Time Systems and Applications
Summer Semester 16

Exercise Sheet 2

(11 Punkte)

Exercise Due at Wednesday, April 27, 2016, 13:00 Uhr

Hinweise: Gruppenarbeit von bis zu drei Personen aus der gleichen Übungsgruppe ist möglich. Bitte vergessen Sie nicht Ihre Namen und Ihre Matrikelnummern auf die Lösung zu schreiben. **Die Abgaben können in den beschrifteten Briefkasten vor dem Sekretariat des LS12 (OH16/E22) eingeworfen oder per Mail (PDF Format) an georg.von-der-brueggen [☺] tu-dortmund.de abgegeben werden.**

Note: It is allowed to work in a group of up to three persons, if these persons are from the same practice group. Please do not forget to write your name and your Matrikelnummer on the solutions. **The solutions can either be placed in the mailbox in front of the secretary's office of LS 12 (OH/E22) or sent by mail (PDF format) to georg.von-der-brueggen [☺] tu-dortmund.de**

Exercise Sessions:

Do, 10:15 - 11:45	OH16/U08
Do, 14:15 - 15:45	OH16/U08

2.1 RM Scheduling (2 Punkte)

Suppose that we are given the following 3 sporadic real-time tasks with implicit deadlines.

	τ_1	τ_2	τ_3
C_i	1	2	3
T_i	4	6	10

1. What are their priority levels? Is the rate-monotonic (RM) schedule feasible?
2. What happens if we change the minimum inter-arrival time of task τ_3 from 10 to 8?

2.2 Critical Instant Theorem (3 Punkte)

Explain the critical instant theorem for uniprocessor fixed-priority scheduling in your words. As mentioned in the lecture, the critical instant theorem for uniprocessor fixed-priority scheduling is very fragile if the assumptions are not met. To apply the critical instant theorem, quite a few conditions have to be satisfied. Please indicate which of the following conditions are correct and which of them are incorrect. If a condition is incorrect, please correct it.

- The task set consists of only independent tasks.
- The task set must be *strictly* periodic.
- The scheduling algorithm is fixed-priority preemptive scheduling.
- Early completion of jobs is not possible. A job has to spin till its worst-case execution time if it finishes earlier.
- No task voluntarily suspends itself. That is, a job cannot suspend itself during its execution.
- The relative deadline of a task can be larger than its period.
- Scheduling overheads (context switch overheads) are zero.
- All periodic/sporadic tasks have zero release jitter (the time from the task arriving to it becoming ready to execute).

2.3 Optimality of RM (2 Punkte)

Explain how to use the time-demand analysis (TDA) to prove that rate-monotonic scheduling is an optimal static-priority scheduling policy (with respect to schedulability) for task systems with implicit deadlines.

Hint: You can prove by swapping two adjacent tasks (in the priority order) if they do not follow the RM scheduling policy. As long as the schedule after swapping the priority levels of these two tasks remains feasible, we can keep swapping the tasks to convert the order into RM such that the RM schedule remains feasible.

2.4 Harmonic Task Systems (3 Punkte)

Given a set $\mathcal{T}$ of n independent, preemptable, and periodic tasks with implicit deadlines, they can be partitioned into $\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_k$ task sets, in which $\bigcup_{j=1}^{k} \mathcal{T}_j$ is $\mathcal{T}$. Moreover, for $j = 1, 2, \ldots, k$, the periods of the tasks in each task set $\mathcal{T}_j$ are *simply periodic* or *harmonic*. That is, for any $\tau_i, \tau_\ell \in \mathcal{T}_j$, $\frac{T_i}{T_\ell}$ is a positive integer if $T_i \geq T_\ell$.

Prove that if

$$\sum_{i=1}^{n} U_i \leq k(2^{\frac{1}{k}} - 1),$$

rate-monotonic scheduling is a feasible schedule.

Hint: Convert these n tasks to a more difficult case with only k tasks.

2.5 Challenge (Optional Exercise with Bonus) (1 Punkt)

Mr. Smart suggests the following schedulability test of static-priority scheduling for sporadic real-time tasks, as defined in the course. He claims that task τ_i can meet its its relative deadline under the static-priority scheduling if and only if the following mixed-integer linear programming has a solution.

$$C_i + \sum_{j=1}^{i-1} n_j \cdot C_j \leq t \tag{1}$$

$$n_j \cdot T_j \geq t \qquad \forall j = 1, 2, \ldots, i-1 \tag{2}$$

$$n_j \in \mathbf{N} \qquad \forall j = 1, 2, \ldots, i-1 \tag{3}$$

$$0 < t \leq D_i, \tag{4}$$

where t is a positive variable, described in (4), and n_j is a positive integer number, described in (3). Please either explain/prove or disprove his argument.

General Hints: