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

Exercises for Lecture
Real-Time Systems and Applications
Summer Semster 15

Exercise Sheet 7

(11 Punkte)

Exercise Due at Wednesday, June 24, 2015, 12:00 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/E18
Do, 14:15 - 15:45	OH16/E18

Definitions: Assume the tasks $\tau_i \in \tau$ are ordered according to Rate Monotonic and that all tasks have different periods, i.e. $T_i \neq T_j \; \forall \tau_i, \tau_j \in \tau$ with $i \neq j$

- $hp(\tau_k)$ consists of the $\tau_i \in \tau$ with $T_i < T_k$, i.e., all tasks with higher priority then τ_k
- $lp(\tau_k)$ consists of the $\tau_i \in \tau$ with $T_i > T_k$, i.e., all tasks with lower priority then τ_k
- Blocking time: $B_k = \max_{\tau_i \in lp(\tau_k)} \{C_i\}$
- t_k is the deadline of T_k
- $\{t_1, \ldots, t_k\}$ are the last releases of the tasks $\tau_i \in hp(\tau_k)$ before t_k and t_k itself

7.1 k^2U Framework (4 Punkte)

1. Use the k^2U Framework to prove the Liu and Layland Bound for implicit deadline task sets under Rate Monotonic Scheduling

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

2. A sufficient schedulability test for non-preemptive rate monotonic scheduling is to show, that $\exists t_j \in \{t_1, \ldots, t_k\}$ such that

$$C_k + B_k + \sum_{i=1}^{k-1} \left\lceil \frac{t_j}{T_i} \right\rceil C_i \leq t_j$$

if the schedulability of all higher priority tasks was already guaranteed.

Show that a task τ_k in a non-preemptive sporadic task system with implicit deadlines can be feasibly scheduled by the Rate Monotonic scheduling algorithm, if the schedulability for all higher priority tasks has already been ensured and the following condition holds:

$$\left(\frac{C_k + B_k}{T_k} + 1\right) \prod_{\tau_j \in hp(\tau_k)} (U_j + 1) \leq 2$$

Hint: Split the summation into two parts, one representing all releases of the higher priority tasks before the last release and the other one representing the last release.

Hint: A sufficient test for preemptive rate monotonic scheduling is to use the test presented for the non-preemptive case and remove the blocking time.

7.2 Multi-Frame Task Systems (4 Punkte)

Given n independent, preemptable, and periodic tasks with implicit deadlines, there are two types of executions for each task τ_i. It is known that $C_{i,2} \leq C_{i,1} \leq 2C_{i,2}$, in which $C_{i,1}$ is the WCET of the first version, and $C_{i,2}$ is the WCET of the second version.

Suppose that the tasks are ordered in a rate-monotonic order, in which $T_i \leq T_{i+1}$ $i = 1, 2, \ldots, n-1$ and $2T_1 \geq T_n$. For any two consecutive jobs of task τ_i, at most one of them is executed by using the first-version (i.e., $C_{i,1}$).

The task set is schedulable under rate monotonic scheduling if

$$\prod_{i=1}^{n} \left(\frac{C_{i,2}}{T_i} + 1 \right) \leq 1.5.$$

Prove the above statement, and also generalize the above bound for another condition $C_{i,2} \leq C_{i,1} \leq \alpha C_{i,2}$ where $\alpha \geq 1$.

Hint: The critical instant theorem for this case is as follows: If the worst-case response time of task τ_k is no more than T_k, then, the response time of a job of task τ_k, arriving at time t, by releasing all the higher-priority tasks τ_is with execution time $C_{i,1}$ at time t and the subsequent job with execution time $C_{i,2}$ at time $t + T_i$ as early as possible by respecting the period, is the worst-case response time of task τ_k.

Hint: Write down the schedulability test and use the k^2U framework introduced in the lecture.

7.3 Self-Suspension Modeling and Scheduling (2 Punkte)

- Give concrete examples why EDF and RM can be very bad when considering tasks with self-suspensions.
- Define the dynamic self-suspension and the segmented self-suspension task models for sporadic real-time tasks. What are their advantages and disadvantages with respect to the expressiveness of the system and the accuracy in the schedulability design/analysis, respectively?

7.4 Challenge on Utilization Bounds (1 Punkt)

Based on 7.1-2 prove the following argument.

Suppose that the tasks are indexed such that $T_i \leq T_{i+1}$. If $\gamma = \max_{\tau_i \in lp(\tau_k)} \left\{ \frac{C_i}{C_k} \right\} = \frac{B_k}{C_k}$, then task τ_k is schedulable by RM-NP if

$$\sum_{i=1}^{k} U_i \leq \begin{cases} \left(\left(\frac{2}{\gamma}\right)^{\frac{1}{k}} - \frac{1}{1+\gamma} \right) + (k-1)\left(\left(\frac{2}{\gamma}\right)^{\frac{1}{k}} - 1 \right) & \text{if } \gamma \leq 1 \\ \frac{1}{1+\gamma} & \text{if } \gamma > 1 \end{cases}$$