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

Exercises for Lecture
Real-Time Systems and Applications
Summer Semester 17

Exercise Sheet 7

(11 Punkte)

Exercise Due at Thursday, July 6, 2017, 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 gesendet 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 marked mailbox in front of the secretary's office of LS 12 (OH16/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

7.1 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?

7.2 Self-Suspension (3 Punkte)

Prove that the speedup factor of the *Proportional Fixed-Relative Deadline Assignment* for 1-segmented self suspension sporadic task systems is at least $\Omega(n)$. The assignment is

- $D_{i,1} = \frac{C_{i,1}}{C_{i,1}+C_{i,2}}(T_i - S_i)$
- $D_{i,2} = \frac{C_{i,2}}{C_{i,1}+C_{i,2}}(T_i - S_i)$

Hint: Consider the following task set with n tasks where $n \geq 5$:

- $C_{i,1} = 1, C_{i,2} = 2^i - 1$,
- $T_i - S_i = 2^i * (n-1)$, and
- $T_i = F > 2^n * (n-1)$
- Therefore, $D_{i,1}$ is set to $n-1$, and $D_{i,2}$ is set to $(2^i - 1) * (n-1)$

You can assume that this task set can be feasibly scheduled by using equal deadline assignment introduced in the lecture.

7.3 Necessary Condition and Sufficient Condition (2 Punkte)

Consider task systems with constrained deadlines.

- Write down a necessary condition for dynamic self-suspension sporadic task systems under uniprocessor fixed-priority scheduling.
- Write down a sufficient condition for dynamic self-suspension sporadic task systems under uniprocessor fixed-priority scheduling.

Explain/prove your answer.

7.4 Self-Suspension due to Shared Resources (3 Punkte)

Consider an implicit-deadline task set $\mathbf{T}$. Each task τ_i in $\mathbf{T}$ is assigned to run only on its designated processor has

- C_i as worst-case execution time on its assigned processor (which ignores shared resource access)
- A_i as worst-case execution time on a physical shared resource, e.g., bus, (which ignores computation on its assigned processor).

This means, $C_i + A_i$ is the worst-case execution time of task τ_i if there is no contention on the physical shared resource. Suppose that there is only one physical shared resource. Both shared resource accesses and processor executions are based on preemptive RM fixed-priority scheduling. Prove and explain that the task set is schedulable under the scheduling policy if

$$\forall \tau_k \in \mathbf{T}, \exists t | 0 < t \leq T_k \text{ s. t. } \quad \left(C_k + \sum_{\tau_i \in hp(\tau_k, c)} \left\lceil \frac{t + T_i}{T_i} \right\rceil C_i \right) + \left(A_k + \sum_{\tau_i \in hp(\tau_k, r)} \left\lceil \frac{t + T_i}{T_i} \right\rceil A_i \right) \leq t,$$

where

- $hp(\tau_k, c)$: higher-priority tasks than τ_k assigned to the same core as task τ_k, and
- $hp(\tau_k, r)$: higher-priority tasks than τ_k on the shared resource.

7.5 Challenge on Arbitrary Schedules with 1-Segmented Self-Suspension (optional) (1 Punkt)

Prove the following necessary condition for any implicit-deadline arbitrary 1-segmented self-suspension sporadic task model:

$$\forall t > 0, \qquad \sum_{\tau_i \in \mathbf{T}} dbf_i^*(t) \leq t. \tag{1}$$

where

$$dbf_i^*(t) = \begin{cases} 0 & 0 \leq t < T_i - S_i \\ C_{i,\max} + \left\lfloor \frac{t-(T_i-S_i)}{T_i} \right\rfloor (C_{i,1} + C_{i,2}) & t \geq T_i - S_i \end{cases} \tag{2}$$

General Hints: