Prof. Bernd Finkbeiner, Ph.D.
Dipl.-Inf. Rüdiger Ehlers
Markus Rabe, M.Sc.
Sebastian Hahn, B.Sc.

Summer term 2012
Problem Set 9
Due: Wednesday, 27^{th} June 2012

Embedded Systems

Please indicate your **name**, **matriculation number**, **email address**, and which **discussion session** you have been allocated to. We encourage you to collaborate in **groups** of up to **three** students. Only one submission per group is necessary.

Exercise 1: Periodic Scheduling

For each of the following tasks sets, (1) determine whether an EDF-schedule and/or an RM-schedule exists, and (2) formally prove your answer.

$\Gamma = \{\tau_1, \tau_2, \tau_3\}$	$T_1 = D_1 = 3$	$C_1 = 1$
	$T_2 = D_2 = 4$	$C_2 = 2$
	$T_3 = D_3 = 8$	$C_3 = 1$
$\Delta = \{\tau_1, \tau_2, \tau_3\}$	$T_1 = D_1 = 2$	$C_1 = 1$
	$T_2 = D_2 = 3$	$C_2 = 1$
	$T_3 = D_3 = 4$	$C_3 = 1$
$\Pi = \{\tau_1, \tau_2, \tau_3, \tau_4\}$	$T_1 = D_1 = 2$	$C_1 = 1$
	$T_2 = D_2 = 5$	$C_2 = 1$
	$T_3 = D_3 = 8$	$C_3 = 2$
	$T_4 = D_4 = 20$	$C_4 = 1$

Exercise 2: Aperiodic Scheduling

Consider the following scheduling problem $1 \mid sync \mid T_w$:

> Using a uniprocessor machine, find a schedule for a set $\mathcal{J} = \{J_1, \ldots, J_n\}$ of n synchronous tasks with computation times $C_1, \ldots, C_n$ that minimizes the weighted sum of the completion times
>
> $$T_w = \sum_{i=1}^{n} (w_i f_i)\,,$$
>
> where $w_i > 0$ is a weight, and f_i is the time at which task i finishes its execution. (*Note:* The schedule is not required to respect the deadlines. We are only interested in minimizing T_w.)

(a) Let $\mathcal{J}$ be a task set, and let σ be a schedule for $\mathcal{J}$ that is optimal with respect to the problem $1 \mid sync \mid T_w$. Formally prove that there exists a nonpreemptive schedule σ^* for $\mathcal{J}$ with the same T_w of σ.

(b) Devise a polynomial-time algorithm that, given a task set $\mathcal{J} = \{J_1, \ldots, J_n\}$, computes a schedule σ for $\mathcal{J}$ that is optimal with respect to the scheduling problem $1 \mid sync \mid T_w$.

(c) Formally prove that your algorithm computes an optimal schedule.

Exercise 3: Optimality of Aperiodic scheduling

Consider the problem of scheduling a set of synchronous tasks on a uniprocessor machine. It was shown in class that Jackson's EDD algorithm minimizes the maximum lateness

$$L_{max} = \max_i (f_i - d_i)\,.$$

For each of the following criteria, determine whether the EDD algorithm minimizes it:

(a) average response time $\overline{R} = \frac{1}{n}\sum_{i=1}^{n} f_i$;

(b) total completion time $T_c = \max_i(f_i)$;

(c) weighted sum of completion times $T_w = \sum_{i=1}^{n} w_i f_i$;

(d) number of late tasks $N_{late} = \sum_{i=1}^{n} (\mathit{if}\ \ d_i > f_i\ \ \mathit{then}\ 1\ \mathit{else}\ 0)$.

For each criterion, if EDD minimizes it, give a formal proof; otherwise give a counterexample.