IA158 Real Time Systems 2012
Assignment 1

Instructions

- Deadline is March 26th, 8AM. Deadline is strict.
- Assignment must be delivered on paper (preferably at the lecture, alternatively to my compartment in kitchen B3).
- Assignment may be written by hand (legibly!). Assignment must be in English and signed (name, UCO).
- You must work independently.
- To get the maximum number of points, it is sufficient to solve 4 tasks. If you solve just 4 tasks, you should solve them really properly. If you solve more than 4 tasks, I will tolerate small mistakes or incompleteness of solutions.

Tasks

Task 1. For which values $X, Y \in \mathbb{N}$ is the following set of aperiodic jobs schedulable?

	J_1	J_2	J_3	J_4
a_i	0	1	2	5
C_i	4	X	1	4
d_i	Y	9	8	10

Choose a specific values of X and Y (preferably "optimal" values) and for these values construct a schedule produced by the earliest deadline first algorithm and the least slack time algorithm.

Task 2. Find two sets of 4 periodic tasks with utilization factor $U = 0.95 \pm 0.02$ such that the first task set is schedulable by the rate monotonic algorithm and the second task set is not schedulable by the rate monotonic algorithm. Draw schedules produced by the rate monotonic algorithm.

Task 3. Consider a "Slack time monotonic" (STM) algorithm for scheduling periodic tasks – a fixed priority algorithm which works in a similar way as the rate monotonic algorithm but assigns priorities according to the inverse of the slack time (i.e., $T_i - C_i$) instead of period. We know that RM is optimal among fixed priority algorithms. Show that STM is not optimal among fixed priority algorithms.

Task 4. Design your our example for scheduling of aperiodic jobs with arbitrary arrival times and precedence constraints. The example should contain at least

6 jobs and non-trivial constraints. Solve the example using the transformation of arrival times and deadlines.

Task 5. Design your own example for scheduling with both periodic and aperiodic jobs and construct schedules according to at least three different algorithms (e.g., background scheduling, polling server, deferrable server). The example should be "interesting", i.e., different algorithms should lead to different outcomes.

Task 6. Choose one of the theorems about optimality of scheduling algorithms and prove it. It is not necessary to write a formal proof, but you should at least sketch informally all the main ideas. You may also formulate your own algorithm (for one of the studied problems) and prove its optimality.

Task 7. Choose one of the correct mutual exclusion algorithms and argue that it satisfies all requirements. It is not necessary to write a formal proof, but you should at least sketch informally all the main ideas.