Real-Time Systems (CSE 40463/60463) — Fall 2007

Homework Assignment 1

Due date: September 18, 2007 (in class)

MY NAME IS: ____________________________________

Question 1: Assume that a distributed system consists of four nodes. Their local clocks read: 11550, 11570, 11515, and 11525.
a.) Execute the Berkeley clock synchronization algorithm assuming that the first clock is the time server. What are the new clock readings?
b.) Execute the Cristian clock synchronization algorithm and assume that the returned UTC clock value is 11540 for each clock and that each local clock advances by 6 units while waiting for the time server's response. What are the new clock readings?

Question 2: The following tables show three sets of periodic tasks. For each set, indicate if the system is "overloaded" and justify your answer.

Table 1: Task Set I

Task i	**Period** P_i **(ms)**	WCET e_i (ms)
1	10	4
2	7	1
3	9	5

Table 2: Task Set II

Task i	**Period** P_i **(ms)**	WCET e_i (ms)
1	3	1
2	5	2
3	11	1
4	13	1
5	15	1

Table 3: Task Set III

Task i	**Period** P_i **(ms)**	WCET e_i (ms)
1	15	3
2	18	5
3	23	3

Question 3: Assume you have three periodic tasks: $P_1(8,2)$, $P_2(16,5)$, and $P_3(32,6)$, where $P(p,e)$ indicates a periodic task with period p and worst-case execution time e. Generate a feasible schedule using the offline scheduling approach. Show exactly the amount of information needed to fill a scheduling table.

Question 4: Consider the following periodic task set: P_1(5,1), P_2(10,1), P_3(15,2), and P_4(15,1), where $P(p,e)$ indicates a periodic task with period p and worst-case execution time e. At time 4 a sporadic task $S_1(18,3)$ arrives, at time 7 sporadic task $S_2(37,3)$ arrives, at time 13 sporadic task $S_3(26,3)$ arrives, and at time 16 sporadic task $S_4(53,11)$ arrives. Draw a schedule to explain which sporadic tasks are accepted and which are rejected and clearly indicate the time each accepted sporadic task finishes its execution.

Programming Assignment 1 (DUE: 9/25): Write a simulator (in your favorite programming language) that generates schedules for the following scheduling algorithms: RMS, EDF, LST. Your program allows a user to provide any number of periodic tasks, e.g., via command line parameters, text input, or file input. Another input parameter is the size of the schedule. Your program will print a schedule (to screen or file), CLEARLY indicating when a job is being preempted and when a job misses its deadline. Your output will also include a summary, indicating the average response time for each task and the number of preemptions and deadline misses per task and in total. Submit the source code and binary (plus any other files necessary for compilation such as a makefile) in the AFS dropbox. Further include at least 3 examples (task set plus corresponding schedule) with your submission and provide a brief report (at most 1 page) discussing the differences in the per-task response times and deadline misses. If necessary (e.g., compilation difficulties), the instructor may ask you to come to an office hour to discuss your submission.