Problem Set 2

UBC | EECE 494 | Spring 2011

Assigned: March 25
Due: April 8

Each question is worth 10 points. If a question has multiple parts then each part is equally weighted.

1. A single on-board computer controls several features on a new car. The on-board computer plays music from a local hard-disk, and this task (Task T_1) requires 20ms of time every 100ms. The GPS system (Task T_2) that provides directions requires 30ms of execution time every 250ms. The one other task that runs on this computer manages the temperature and humidity in the car; this task (Task T_3) is performed every 400ms and requires 100ms of execution each period. Tasks are scheduled using the **rate monotone** priority assignment.

   (a) You are a new engineer on this project and you have been assigned the task of integrating an additional feature on this on-board computer. This feature, a traffic monitor, scans a special communications channel and identifies routes that are congested or under repair. The traffic monitor (Task T_4), as it is currently implemented, runs for 100ms every 280ms. You need to determine if this task can be introduced without causing any deadline violations. If not, you need to instruct the feature engineers to redesign the traffic monitor and reduce its execution time (you cannot alter the frequency because it has been determined to provide drivers with sufficient time to change routes). What would your recommendation be?

   (b) Assume that only the original three tasks (T_1, T_2, T_3) are running on the on-board computer. In a redesign stage, it is determined that these tasks need to update a display by sending some information over a data bus. This communication takes time but the display needs to be updated within the task's period. As a result, the relative deadlines for the three tasks need to be shortened and the tasks scheduled using the **deadline monotone** priority assignment. For simplicity, all tasks will have their deadlines reduced by a factor f. In other words, the relative deadline D_i for task T_i will become $D_i = f \cdot P_i$ where P_i is the period of the task. What is the smallest value of f such that the tasks will continue to meet their deadlines. [Do not worry about issues such as blocking and mutual exclusion.]

2. Consider a task set with three periodic tasks: $\tau_1 = (P_1 = 7, e_1 = 1), \tau_2 = (14, 2), \tau_3 = (28, 3)$. Assume that all tasks release a job at time $t = 0$. An aperiodic job arrives at $t = 5$ and has an execution time requirement of 2. This aperiodic job is followed by another aperiodic job that releases at time $t = 9$ and requires 3 units of execution. Using the following policies and servers, determine the completion time of the two aperiodic jobs.

   (a) Rate monotonic scheduling with a polling server of period 7 and budget 2.

   (b) Rate monotonic scheduling with a sporadic server of period of period 7 and budget 2.

3. Consider a processor P in an end-to-end computing system that uses the *release-guard protocol* to synchronize subtasks on different processors. Only two tasks execute on P. P is the j^{th} stage for task T_1 and $T_{1,j} = (P_1 = 4, e_{1,j} = 2)$. P is also the first stage of execution for task T_2 with $T_{2,1} = (P_2 = 10, e_{2,1} = 4)$. The first three synchronization signals from stage $j-1$ of task T_1 come at times $t = 1, 5$ and 7 respectively. Jobs on P are scheduled rate monotonically. So, when are the first three jobs of T_1 released on P?

4. Let us suppose that there are certain tasks that need to be executed periodically but an additional requirement stipulates that for any task T_i, with period P_i, successive instances must not complete more than P_i time units apart.

   (a) Is the standard rate monotonic scheduling policy sufficient in order to satisfy this requirement? Explain your answer.

   (b) How can a task be transformed such that the RM scheduling policy will satisfy the constraint on the distance between two successive jobs of a task?

5. Consider the set of three tasks $\{T_1 = (P_1 = 10, e_1 = 2), T_2 = (20, 4), T_3 = (22, 10)\}$. Illustrate the scheduling of these three tasks on a two processor system for the first 45 time units with

   (a) EDF with partitioned scheduling. Tasks are allocated to processors using the first-fit-decreasing heuristic.

   (b) EDF with global scheduling.

6. Three soft real-time tasks have to be scheduled on the same processor. Two tasks have fixed execution times but their periods/frequencies are flexible. The third task has a fixed periodicity but variable execution times because of supporting imprecise execution. The parameters of the tasks are as in the following table (Table 1).

   k is the elasticity of each task and e_0 and P_0 are the *nominal* execution times and nominal periods for the tasks. Assume that elastic scheduling is used.

Task	e_{min}	e_{max}	e_0	P_{min}	P_{max}	P_0	k
T_1	2	2	2	8	12	10	1
T_2	4	4	4	5	10	6	2
T_3	4	10	5	12	12	12	1.5

Table 1: Task set for question 6

What then are the execution parameters of these three tasks? Recall that the formula for elastic scheduling is as follows:

$$\forall T_i \in V : U_i = U_{i,0} - (U_0 - U_b + U_M)\frac{k_i}{\sum_{T_j \in V} k_j}.$$

Solve the problem for both EDF scheduling and RM scheduling using the Liu and Layland bound.

7. Consider a set of flexible tasks scheduled using the deadline monotonic scheduling policy. The utilization of each task T_i is fixed at u_i but the tasks are flexible because different instances of a task may have different execution times and relative deadlines subject to the task's utilization. In other words, $T_{i,j}$, the j^{th} instance of T_i will have execution time $e_{i,j}$ and relative deadline $D_{i,j}$ such that $u_i = e_{i,j}/D_{i,j}$. A new instance of a task is released at the deadline of the previous instance of the same task. This model is similar to the strictly periodic task model but does the Liu and Layland bound apply to this model? Explain your answer with an example.

8. A fault-tolerant computing structure F consists of identical elements, each with a constant failure rate λ. The outputs of these two elements are compared; a mismatch invokes a diagnostic routine that can identify the faulty element correctly with probability c. The diagnostic test leads to the shutdown of the element identified as malfunctioning; after this step there is no more output comparison. In this setup, what is the reliability of the structure? (The system is functioning correctly if at least one element is operating without a fault.)