Problem Set #1

UBC | EECE 494 | Spring 2010

February 8, 2010

Due: March 3, 2010

This problem set is an individual assignment. You are allowed to consult lecture notes, textbooks and other such material. You may discuss the questions with other persons, particularly the instructor and the teaching assistant, however all the work that you hand in must be your own. If you benefited from discussions with other persons, you must mention that person's name. If you used resources other than the lecture notes and the readings on the course website, indicate the source of the material. **Plagiarism will be dealt with severely.**

It is recommended that you add a cover page to your solutions that clearly indicates your name, your student number, and the sources that you consulted and the names of other persons with whom you discussed the problem set.

You can get a maximum of 75 marks on this problem set, which is worth 5% of the total course grade.

1. Indicate if the following statements are *true* or *false*. **[1 point each]**

   (a) `POSIX` implements several scheduling policies.

   (b) It is possible to model strictly non-preemptive execution as blocking time in a preemptive scheduling context.

   (c) Unbounded blocking and deadlocks can be avoided by using the priority inheritance protocol.

   (d) Aperiodic task servers help aperiodic tasks meet deadlines.

   (e) `POSIX` signals are primitives that permit condition synchronization.

2. Consider two periodic tasks $(20, 15)$ and $(40, 10)$ that need to be executed on the same processor.

   (a) Assume the tasks are scheduled using the round-robin policy where each task is allocated one unit of processor time and is then preempted to allow the next task to execute. Show that a feasible schedule *does not* exist. **[3 points]**

   (b) Assume that the tasks are scheduled using the *shortest job first* policy. Does a feasible schedule exist? Show the existence or non-existence of a feasible schedule. **[3 points]**

3. Consider the following task set:

Task	e_i	D_i	P_i
T_1	2	5	6
T_2	2	4	8
T_3	4	8	12

   (a) What is the worst-case response time for task T_3? **[3 points]**

   (b) If this task set were scheduled using the earliest deadline first policy then indicate the smallest set of time instants at which one would need to test the processor demand criterion before we can conclude that the task set is schedulable using EDF? **[3 points]**

4. Use response time analysis to verify if the following task set (with known blocking times) is schedulable using the rate monotonic scheduling policy. (One blocking term is not specified. What should it be?) **[3 points]**

Task	e_i	P_i	B_i
T_1	4	10	5
T_2	3	15	3
T_3	4	20	?

5. State two advantages of the priority ceiling protocol *with immediate inheritance* when compared to the basic priority inheritance protocol. **[3 points]**

6. Consider a real-time system that utilizes a polling server with period P and budget C to accommodate aperiodic tasks. What is the worst-case response time for an aperiodic job that requires $2C$ units of computation, assuming no aperiodic jobs were pending at the time this job arrived and no aperiodic jobs arrived until this job was completed? You need to compute this worst-case response time when tasks are scheduled rate monotonically and the server is the highest priority task in the system. **[4 points]**

7. Consider the maximum utilization first static priority policy that prioritizes tasks with higher utilization, i.e., greater the value of e_i/P_i the greater the priority of the task. Obtain a utilization bound, U_b, for this scheduling policy when all tasks are periodic and have their period equal to their relative deadline. Whenever a task set satisfies this bound it must be schedulable; even if it does not satisfy the bound then it may be schedulable. Your bound, U_b, should be tight, i.e., there must exist at least one task set with utilization $U_b + \varepsilon$ for any $\varepsilon > 0$ that will miss deadlines. What is the limiting case when the number of tasks is very large? Show your steps. **[12 points]**

8. Prove the following proposition for a set of periodic tasks: *For a task τ_i, with period P_i, execution time e_i, relative deadline D_i, when all tasks are scheduled using deadline monotonic priorities, define $S_i = \{s_{i,k}\}$ to be as follows*

$$\begin{aligned} S'_i &:= \{lP_j \mid j = 1,2,\ldots,i; l = 1,\ldots,\lfloor \frac{P_i}{P_j} \rfloor\}, \\ S_i &:= \{S'_i - \{t \mid t \in S'_i, t > D_i\}\} \cup \{D_i\}. \end{aligned}$$

(By the critical instant theorem, there is no loss of generality in assuming that all tasks release a job at time 0.)

Assume S_i is sorted in ascending order, and that $j < i$ implies τ_j has greater priority than τ_i. τ_i can meet its deadlines if there exist one or more time points $s_{i,k} \in S_i$ that satisfy

$$\sum_{j=1}^{i} e_j \lceil \frac{s_{i,k}}{P_j} \rceil \leq s_{i,k}.$$

[12 points]

9. In a periodic task system with relative deadlines equal to the periods, rate monotonic scheduling may do as well as the earliest deadline first policy if there is a specifici relationship among the task periods. If there are n tasks to be scheduled and the periods of the tasks are $P, kP, k^2P, \ldots, k^{n-1}P$ respectively (where k is an integer greater than 1) then the utilization bound is 1 and not $n(2^{1/n} - 1)$. For example, if there are four tasks and their periods are $4, 8, 16, 32$ respectively then the utilization bound is 1. In fact, when a set of tasks have periods that follow this pattern (called a *harmonic sequence*) then that set of tasks can be replaced by one virtual task for the purpose of analysis. Using this idea, develop a procedure for testing schedulability using utilization bounds giving some attention to the possibility of a harmonic chain existing. Provide a general method **[8 points]** and then use it to test the schedulability of the following task sets **[2 points each]**:

(a) $(e_1 = 1.5, P_1 = 6), (2,12), (2,18), (1,24), (4,48), (6,54), (4,96)$.

(b) $(e_1 = 1, P_1 = 5), (1,10), (1,15), (2,20), (1,25), (3,30), (2,40), (5,75), (2,80), (5,225)$.

10. Determine if the following task set can be scheduled using the rate monotonic scheduling policy with the priority ceiling protocol to control resource access.

Task	e_i	P_i	Resources used
T_1	4	10	R_1, R_2
T_2	5	20	R_2, R_3
T_3	10	35	R_3
T_4	2	40	R_1

The duration for which each resource is used by the tasks is specified in the following table. You may assume that a task locks only one resource at a time.

Resource	Duration
R_1	2
R_2	1
R_3	2

11. **[4 points each]** You should consult the readings to answer these questions fully.

(a) Stankovic, in his article on misconceptions about real-time computing, talks about the serializability of databases. What is serializability? Explain with an example.

(b) How does EDF behave when the system is in a state of permanent overload (utilization > 1)? Which tasks miss deadlines? Compare this situation with the behaviour of RM scheduling under conditions of permanent overload.

(c) How can lock-free synchronization schemes help in a real-time system? Provide a brief discussion that includes schedulability analysis issues.