Written Assignment #2

CPEN 432: Real-time System Design

---

1. **[10 points]** (Yet another exact schedulability test) Consider a constrained-deadline periodic task system consisting of n tasks $T_1, \ldots, T_n$, where task T_i is characterized by a period P_i, a relative deadline $D_i \leqslant P_i$, and a WCET $C_i > 0$. All tasks arrive at time 0. Suppose that the tasks are scheduled on one processor using the **deadline-monotone** policy. Assume for simplicity that the tasks are ordered by decreasing priorities: T_i has higher priority than T_j iff $i < j$. Let

$$\delta_i(t) = \sum_{j=1}^{i-1} \left\lceil \frac{t}{P_j} \right\rceil C_j + C_i \qquad (0 < t \leqslant P_i).$$

Let $\Delta_i(t) = \delta_i(t)/t$.

   (a) **[3 points]** Show that T_i is schedulable if and only if $\min_{(0,D_i]} \Delta_i(t) \leqslant 1$. Conclude that the entire task set is feasible if and only if $\max_{i\in\{1,\ldots,n\}} \min_{(0,D_i]} \Delta_i(t) \leqslant 1$

   (b) **[2 points]** Give an interpretation of this test from an economics perspective.

   (c) **[3 points]** Show that in order to deem task T_i schedulable, it is sufficient that $\Delta_i(t) \leqslant 1$ for *some* point in the testing set

$$S_i = \Big\{ kP_j : j \in \{1, \ldots, i-1\}, k \in \big\{1, \ldots, \lfloor \min(P_i, D_i)/P_j \rfloor\big\}\Big\}.$$

   (d) **[2 points]** From the previous part, derive the asymptotic running time of the schedulability test for the entire task set. Is this a polynomial-time test? Justify your answer.

2. **[10 points]** (Knowledge is Power) 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 specific 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 and then use it to test the schedulability of the following task sets:

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

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

   Note: The description of the algorithm and the proof of its correctness is worth **6 points**, and the calculation in each part is worth **2 points**.

3. **[10 points]** Fix a static-priority scheduling policy. Let Γ denote an implicit-deadline periodic task set, and let $U(\Gamma)$ be its utilization factor. Let

$$b_n = \inf\{U(\Gamma) : \Gamma \text{ is a critically schedulable set of } n \text{ tasks}\}.$$

That is, b_n is the utilization bound for the scheduling algorithm on n tasks (e.g., $b_n = n(2^{1/n} - 1)$ in the LL bound for RM). Let U_n denote the utilization factor of a set of n tasks. *Show that if b_n is a decreasing sequence, then for any set of n tasks with utilization factor U_n, if $U_n \leqslant b_n$, then the taskset is schedulable under the fixed-priority scheme.*

We took this for granted when we derived the utilization bound for RM in class, but this—the sufficiency of the bound—is not so obvious and needs proof.

4. **[10 points]** You need to implement a real-time application with three periodic tasks. Task T_1 has a period of 100ms, relative deadline of 80ms and execution time of 10ms; task T_2 has a period of 75ms, relative deadline of 50ms and execution time of 15ms; task T_3 has a period of 150ms, relative deadline of 100ms and execution time of 30ms. Luckily for you, the operating system supports EDF scheduling and hence you can get better processor utilization. On the other hand, you would like to conserve power and run the processor at the lowest possible speed. If the timing information was obtained using a 500MHz processor, what is the lowest processor speed that would ensure that all jobs meet their deadlines? Assume that execution times scale linearly with speed.

5. **[10 points]** (Accordingly, the poet should prefer probable impossibilities to improbable possibilities.) Consider a set of n independent, implicit-deadline *sporadic* tasks. We do not have exact period and WCET estimates for each task; instead, task periods and execution demands are described by probability distributions. The period of task τ_i is a real-valued random variable P_i that is uniformly distributed in the interval $[i + 4, i + 7]$, and the execution time (demand) of task τ_i is a real-valued random variable C_i that is uniformly distributed in $[i, i + 3]$. Thus task τ_i's utilization is the random variable C_i/P_i. All task execution times and periods are independent.

   (a) **[4 points]** What is the *expected* utilization factor of this task set? Derive a closed form expression for $\mathbb{E}\sum_{i=1}^{n} U_i$ that involves only task indexes, where $\mathbb{E}$ is the expectation operator.

   (b) **[6 points]** Suppose you know only the expected utilization of the taskset, and you want to explore the adequacy of the expected utilization if it were to be used to analyze timing behavior. Typically, the "closer" the expected utilization to the actual utilization realized at run-time, the more representative it is of the actual workload. Show that for given $\varepsilon > 0$ and $0 < \delta \leqslant 1$, if the number of tasks n is $O\left(\varepsilon^2/\log(1/\delta)\right)$, then the probability of the event "*the actual taskset's utilization factor exceeds the expected utilization factor by ε*" is at most δ. Based on this, is it better to use the expected utilization when the number of tasks is large or small? Justify your answer. **Hint:** Read about *concentration inequalities*.