IN4343 – Real Time Systems
April 11th 2019, from 09:00 to 12:00

Mitra Nasri

<table>
<thead>
<tr>
<th>Question:</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Points:</td>
<td>6</td>
<td>15</td>
<td>5</td>
<td>2</td>
<td>28</td>
</tr>
<tr>
<td>Score:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- This is a closed book exam
- You may use a simple calculator only (i.e. graphical calculators are not permitted)
- Write your answers with a black or blue pen, not with a pencil
- Always justify your answers, unless stated otherwise
- Question 8 is an optional question (bonus).

- The exam covers the following material:
  (b) the paper “The Worst-Case Execution-Time Problem” by Wilhelm et al. (except Section 6)
  (c) the paper “Non-Work-Conserving Non-preemptive Scheduling: Motivations, Challenges, and Potential Solutions” by Mitra Nasri and Gerhard Fohler
  (d) the paper “An Exact and Sustainable Analysis of Non-Preemptive Scheduling” by Mitra Nasri and Björn B. Brandenburg
  (e) the paper “Best-case response times and jitter analysis of real-time tasks” by R.J. Bril, E.F.M. Steffens, and W.F.J. Verhaegh
<table>
<thead>
<tr>
<th>Liu and Layland (LL) bound</th>
<th>$U_{lb}^n = n(2^{1/n} - 1)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hyperbolic (HB) bound</td>
<td>$\prod_{i=1}^{n} (U_i + 1) \leq 2$</td>
</tr>
</tbody>
</table>
| Response Time Analysis    | $\begin{align*}
WR_i &= C_i + \sum_{k=1}^{i-1} \left\lfloor \frac{WR_i + AJ_k}{T_k} \right\rfloor \cdot C_k \\
BR_i &= C_i + \sum_{k=1}^{i-1} \left( \left\lfloor \frac{BR_i - AJ_k}{T_k} \right\rfloor - 1 \right)^+ \cdot C_k \\
w^+ &= \max(w, 0)
\end{align*}$ |
| Processor Demand          | $\begin{align*}
g(t_1, t_2) &= \sum_{r_i \geq t_1} C_i \\
g(0, L) &= \sum_{i=1}^{n} \left\lfloor \frac{L + T_i - D_i}{T_i} \right\rfloor \cdot C_i
\end{align*}$ |
| CBS Server                | $\begin{align*}
\text{if (\exists a pending aperiodic job) then} & \text{ enqueue } J_k; \\
\text{else if } (q_s \geq (d_s - a_k) \cdot U_s) & \text{ then} \\
& \{ q_s \leftarrow Q_s; \quad d_s \leftarrow d_s + T_s; \}
\end{align*}$ |
| NP scheduling             | $\begin{align*}
\text{response time NP-FP} & \quad \text{pre-requisites: } D \leq T \text{ and preemptive-schedulable} \\
WO^{(n)}_i &= B_i + \sum_{k=1}^{i-1} \left( \left\lfloor \frac{WO^{(n-1)}_i}{T_k} \right\rfloor + 1 \right) \cdot C_k \\
B_i &= \max\{C_j \mid \forall \tau_j, P_i < P_j\} \\
R_i &= C_i + WO_i \\
WO^{(0)}_i &= B_i + \sum_{k=1}^{i-1} C_k \\
\forall \tau_i, 1 < i \leq n, \forall L, T_1 < L < T_i: & \quad L \geq C_i + \sum_{k=1}^{i-1} \left\lfloor \frac{L - 1}{T_k} \right\rfloor \cdot C_k
\end{align*}$ |
Question 1  

Consider the following code in the context of dynamic techniques to estimate the worst-case execution time of a task on a given platform.

```c
int foo (int x, int y) {
    int z = 0;
    if (x > 7 && y < 5) {
        z = x;
    } else {
        if (x > 4 && y > 0){
            z = 2 * x + y;
        }
    }
    return z;
}
```

(a) **3 points** Generate the minimum number of test cases (i.e., calls to the foo function with different inputs) such that the statement coverage is satisfied.

(b) **3 points** Generate the minimum number of test cases such that the branch coverage is satisfied.

Question 2  

A rate-monotonic scheduler is used to run the following task set.

<table>
<thead>
<tr>
<th>τ</th>
<th>C</th>
<th>T</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>τ₁</td>
<td>3</td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>τ₂</td>
<td>7</td>
<td>18</td>
<td>18</td>
</tr>
<tr>
<td>τ₃</td>
<td>8</td>
<td>24</td>
<td>24</td>
</tr>
</tbody>
</table>

(a) **10 points** Compute the WCRT (worst-case response time) of each task.

(b) **5 points** Determine the worst-case response times of the three tasks. *Hint: draw a time line.*

Question 3  

Park’s test may be used with fixed-priority schedules to verify that the workload to be completed before a task’s deadline is smaller than the deadline of the task itself. The test condition is formalized as:

\[ C_i + \sum_{k=1}^{i-1} \left\lceil \frac{D_i}{T_k} \right\rceil \cdot C_k \leq D_i \]

\(^1\)Thanks to Charles Randolph for preparing this numerical example.

\(^2\)Thanks to Charles Randolph for his suggestion.
(a) [5 points] Is Park’s test a sufficient, necessary, or exact test? Justify your answer.

**Question 4**

(a) [1 point] **Claim:** In general, preemptive fixed-priority scheduling with rate-monotonic priorities has a larger number of preemptions than EDF.
   **Cause:** because each time a high-priority task is released, it will certainly preempt a low-priority task.
   (a) claim is true, cause is true  (b) claim is true, cause is false  (c) claim is false

(b) [1 point] Which resource access policy may result in deadlock?
   (a) Priority Ceiling Protocol (PCP)  (b) Highest-Locker Priority (HLP)
   (c) Priority Inheritance Protocol (PIP)  (d) Non-Preemptive Protocol (NPP)