IN4343 – Real Time Systems
April 7th 2016, from 09:00 to 12:00

Koen Langendoen

<table>
<thead>
<tr>
<th>Question</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Points</td>
<td>10</td>
<td>5</td>
<td>20</td>
<td>15</td>
<td>25</td>
<td>15</td>
<td>90</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

The exam covers the following material:

(a) chapters 1-6, 8-9 of the book “Hard Real-Time Computing Systems (3rd ed)” by G. Buttazzo
(b) the paper “The Worst-Case Execution-Time Problem” by Wilhelm et al. (except Section 6)
(c) the paper “Transforming Execution-Time Boundable Code into Temporally Predictable Code” by P. Puschner
(d) 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
### Liu and Layland (LL) bound

\[ U_{\text{RM}}^{\text{Lub}} = n\left(2^{1/n} - 1\right) \]

### Hyperbolic (HB) bound

\[ \prod_{i=1}^{n}(U_i + 1) \leq 2 \]

### Response Time Analysis

\[ WR_i = C_i + \sum_{k=1}^{i-1} \left( \left\lfloor \frac{WR_k + AJ_k}{T_k} \right\rfloor \right) C_k \]
\[ BR_i = C_i + \sum_{k=1}^{i-1} \left( \left\lfloor \frac{BR_k - AJ_k}{T_k} \right\rfloor - 1 \right) C_k \]
\[ w^+ = \max(w, 0) \]

### Processor Demand

**Schedulability**

\[ \forall L \in D, \quad g(0, L) \leq L \]
\[ D = \{ d_k \mid d_k \leq \min(H, \max(D_{\text{max}}, L^*)) \} \]
\[ H = \text{lcm}(T_1, \ldots, T_n) \]
\[ L^* = \frac{\sum_{i=1}^{n}(T_i - D_i)U_i}{1 - U} \]

**Response Time**

\[ R_a = C_a + \Delta_a + F_a(T_s - C_s) \]
\[ \Delta_a = \left\lceil \frac{r_a}{T_s} \right\rceil T_s - r_a \]
\[ F_a = \left\lceil \frac{C_a - C_0}{C_s} \right\rceil - 1 \]

### Polling Server

**Schedulability**

\[ U_{\text{RM+PS}}^{\text{Lub}} = U_s + n\left(\frac{2}{U_s + 1}\right)^{1/n} - 1 \]
\[ \prod_{i=1}^{n}(U_i + 1) \leq \frac{2}{U_s + 1} \]

**Response Time**

\[ R_a = C_a + \Delta_a + F_a(T_s - C_s) \]
\[ \Delta_a = \left\lceil \frac{r_a}{T_s} \right\rceil T_s - r_a \]
\[ F_a = \left\lceil \frac{C_a - C_0}{C_s} \right\rceil - 1 \]

### Deferrable Server

**Schedulability**

\[ U_{\text{RM+DS}}^{\text{Lub}} = U_s + n\left(\frac{U_s + 2}{2U_s + 1}\right)^{1/n} - 1 \]
\[ \prod_{i=1}^{n}(U_i + 1) \leq \frac{U_s + 2}{2U_s + 1} \]

**Response Time**

\[ R_a = C_a + \Delta_a - C_0 + F_a(T_s - C_s) \]
\[ C_0 = \min(C_s(r_a), \Delta_a) \]
\[ \Delta_a = \left\lceil \frac{r_a}{T_s} \right\rceil T_s - r_a \]
\[ F_a = \left\lceil \frac{C_a - C_0}{C_s} \right\rceil - 1 \]

### NP Scheduling

**Level-i Busy Period**

\[ L_i = B_i + \sum_{h=1}^{i} \left\lceil \frac{L_i}{T_h} \right\rceil C_h \]
\[ B_i = \max_{j>1}\{C_j\} \]

**Response Time**

\[ s_{ik} = B_i + (k-1)C_i + \sum_{h=1}^{i-1} \left( \left\lceil \frac{s_{ik}}{T_h} \right\rceil + 1 \right) C_h \]
\[ R_{ik} = (s_{ik} + C_i) - (k-1)T_i \]
\[ R_i = \max_{k \in [1, N_i]} \{R_{ik}\} \]

### Elastic Model

**Utilization**

\[ \forall i \quad U_i = U_{i0} - (U_0 - U_d) \frac{E_i}{E_S} \quad \text{where} \quad E_S = \sum_{i=1}^{n} E_i \]
Question 1 [10 points]

Determining the worst-case execution time of a task lies at the heart of real-time systems.

(a) 2 points Provide two reasons explaining the variability observed in practice when measuring the execution time of a task.

(b) 3 points Explain why dynamic timing analysis rarely determines the true WCET of a task.

(c) 5 points Explain the concept of scheduling anomalies, and give an example in which knowing the exact WCETs still leads to a deadline violation.

Question 2 [5 points]

Response Time Analysis (RTA) is the key technique to analyze Rate Monotonic scheduling. In the simplest case, when looking at the worst-case response times for tasks without jitter, the RTA equation boils down to:

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

(a) 2 points Argue why it is impossible to derive a closed-form (non-recursive) equation for \( R_i \).

(b) 1 point The RTA equation can be solved by an iterative procedure. Although starting with \( R_i^0 = C_i \) will find the lowest fixed-point solution, one can do better (i.e. fewer iterations) by starting at a higher value. Show that

\[ R_i^* = C_i + \sum_{k=1}^{i-1} \frac{R_i^*}{T_k} C_k \]

is a valid starting point (i.e. \( R_i^* \leq R_i \)).

(c) 2 points Provide a closed-form expression for \( R_i^* \).

Question 3 [20 points]

<table>
<thead>
<tr>
<th>( C_i )</th>
<th>( T_i )</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>5</td>
</tr>
<tr>
<td>2</td>
<td>9</td>
</tr>
<tr>
<td>6</td>
<td>24</td>
</tr>
<tr>
<td>4</td>
<td>48</td>
</tr>
</tbody>
</table>

(a) 10 points Compute the worst-case response times of the tasks under RM scheduling.

(b) 5 points Draw the optimal instant for task \( \tau_3 \), and report its best-case response time.

(c) 2 points Explain why one usually cannot derive the exact response jitter when the worst- and best-case response time of a task are known.

(d) 3 points Compute the bounds on the response jitter of the tasks.
Question 4  [15 points]

<table>
<thead>
<tr>
<th></th>
<th>C_i</th>
<th>D_i</th>
<th>T_i</th>
</tr>
</thead>
<tbody>
<tr>
<td>τ_1</td>
<td>3</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>τ_2</td>
<td>2</td>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td>τ_3</td>
<td>1</td>
<td>9</td>
<td>18</td>
</tr>
</tbody>
</table>

(a) 10 points Determine if the task set is feasible under EDF scheduling.
(b) 5 points Determine if the task set is (still?) feasible under EDF scheduling when relaxing the period of task τ_3 to 20.

Question 5  [25 points]

Although preemptive scheduling is the norm, non-preemptive scheduling has certain advantages that give it an edge in specific conditions. However, it complicates matters as well, reducing schedulability and making analysis more difficult.

Consider the following task set:

<table>
<thead>
<tr>
<th></th>
<th>C_i</th>
<th>T_i</th>
</tr>
</thead>
<tbody>
<tr>
<td>τ_1</td>
<td>3</td>
<td>9</td>
</tr>
<tr>
<td>τ_2</td>
<td>4</td>
<td>12</td>
</tr>
<tr>
<td>τ_3</td>
<td>4</td>
<td>15</td>
</tr>
</tbody>
</table>

(a) 2 points Name two advantages of non-preemptive scheduling.
(b) 3 points Explain why, in the case of RM scheduling, it is no longer sufficient to analyze the critical instant when determining the feasibility of a task set.
(c) 5 points Mention the two conditions under which it suffices to consider only one scheduling period to derive the worst-case response times, and show if they hold for the given task set.
(d) 3 points Explain the notion of level-i busy period and how it can be used to determine the maximum number of periods to derive the worst-case response time of a task.
(e) 12 points Determine the feasibility of the task set under non-preemptive RM scheduling by reporting (i) the level-i busy periods, and (ii) the worst-case response times of the tasks.

Question 6  [15 points]

Consider the set of tasks below:

<table>
<thead>
<tr>
<th></th>
<th>C_i</th>
<th>T_i^{min}</th>
<th>T_i^{max}</th>
<th>E_i</th>
</tr>
</thead>
<tbody>
<tr>
<td>τ_1</td>
<td>5</td>
<td>8</td>
<td>15</td>
<td>1</td>
</tr>
<tr>
<td>τ_2</td>
<td>6</td>
<td>10</td>
<td>20</td>
<td>2</td>
</tr>
<tr>
<td>τ_3</td>
<td>7</td>
<td>12</td>
<td>30</td>
<td>2</td>
</tr>
</tbody>
</table>

(a) 3 points Explain the difference between a transient and permanent overloaded system.
(b) 4 points The Elastic Period Method can be used to derive the periods T_i that would guarantee the schedulability of the given tasks under a specific scheduling policy.
   – Explain the complication of “boundary conditions”, and how to handle them.
(c) 8 points Apply the Elastic Period Method to the above task set for EDF and compute the periods T_i. Assume that all tasks start at their maximum rate (i.e. with T_{min}).