IN4343 – Real Time Systems
April 9th 2014, from 9:00 to 12:00

Koen Langendoen       Marco Zuniga

<table>
<thead>
<tr>
<th>Question</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Points</td>
<td>10</td>
<td>20</td>
<td>20</td>
<td>30</td>
<td>10</td>
<td>90</td>
</tr>
<tr>
<td>Score</td>
<td></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**

- 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

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

### Hyperbolic (HB) bound

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

### Response Time Analysis

- **WR**
  \[ WR_i = C_i + \sum_{k=1}^{i-1} \left\lfloor \frac{WR_k + AJ_k}{T_k} \right\rfloor C_k \]

- **BR**
  \[ BR_i = C_i + \sum_{k=1}^{i-1} \left( \left\lfloor \frac{BR_k - AJ_k}{T_k} \right\rfloor - 1 \right)^+ C_k \]

- \( w^+ = \max(w, 0) \)

### Processor Demand

\[ g(t_1, t_2) = \sum_{r_i \geq t_1} C_i \]

- **schedulability**
  \[ \forall L \in D, \; g(0, L) \leq L \]

- **D**
  \[ D = \{ d_k | d_k \leq \min(H, \max(D_{\text{max}}, L^*)) \} \]

- **H**
  \[ H = \text{lcm}(T_1, \ldots, T_n) \]

- **L^***
  \[ L^* = \sum_{i=1}^{n} (T_i - D_i)U_i \]

### Polling Server

- **schedulability**
  \[ U_{lub}^{\text{RM} + \text{PS}} = U_s + n \left( \frac{2}{U_s + 1} \right)^{\frac{1}{n}} - 1 \]

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

### Deferrable Server

- **schedulability**
  \[ U_{lub}^{\text{RM} + \text{DS}} = U_s + n \left( \frac{U_s + 2}{2U_s + 1} \right)^{\frac{1}{n}} - 1 \]

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

### NP scheduling

- **level-i busy period**
  \[ L_i = B_i + \sum_{h=1}^{i} \left\lfloor \frac{L_h}{T_h} \right\rfloor C_h \]

- \[ B_i = \max_{j>i} \{ C_j \} \]

### 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]

To avoid the intricacies of determining the WCET of a task, the code can be transformed into a single-path equivalent utilizing a predicated execution model. Unfortunately most real-world instruction set architectures only provide a few predicated instructions. As a consequence the transformation process becomes more complex.

```
if (y != 0) {
    norm = x / y;
} else
    norm = 1;
```

(a) 5 points describe the complication(s) with the above code fragment if the only available predicated instruction is a conditional move.

(b) 5 points provide (pseudo) assembly code for the transformed code.

Question 2 [20 points]

Given the following preemptable periodic tasks scheduled based on the Rate Monotonic Algorithm:

<table>
<thead>
<tr>
<th>C_i</th>
<th>T_i</th>
</tr>
</thead>
<tbody>
<tr>
<td>τ_1</td>
<td>2</td>
</tr>
<tr>
<td>τ_2</td>
<td>3</td>
</tr>
<tr>
<td>τ_3</td>
<td>6</td>
</tr>
</tbody>
</table>

(a) 5 points Is there a feasible schedule for these tasks?

(b) 10 points Derive the Best Case Response time for all tasks.

(c) 5 points What is the maximum Activation Jitter we can insert in task τ_1 while still guaranteeing schedulability for the entire task set?

Question 3 [20 points]

Given the following preemptable periodic tasks running under the EDF algorithm:

<table>
<thead>
<tr>
<th>C_i</th>
<th>D_i</th>
<th>T_i</th>
</tr>
</thead>
<tbody>
<tr>
<td>τ_1</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>τ_2</td>
<td>2</td>
<td>5</td>
</tr>
<tr>
<td>τ_3</td>
<td>3</td>
<td>10</td>
</tr>
</tbody>
</table>

(a) 5 points What are the limitations of using the conditions \( \sum_{i=1}^{n} \frac{C_i}{T_i} \) and \( \sum_{i=1}^{n} \frac{D_i}{T_i} \) to check schedulability? Describe the limitations of both conditions.

(b) 10 points Is there a feasible EDF schedule for this set of tasks?

(c) 5 points In general, what conditions—in terms of the \( C_i \)'s, \( D_i \)'s and \( T_i \)'s—would reduce the number of points that need to be checked for schedulability under the processor demand criteria? Describe at least three conditions.
Question 4 [30 points]

When mixing periodic and aperiodic tasks one can make use of a priority server to schedule the aperiodic tasks. Consider the following periodic tasks and aperiodic jobs (under EDF):

<table>
<thead>
<tr>
<th>( C_i )</th>
<th>( T_i )</th>
<th>( a_i )</th>
<th>( C_i )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \tau_1 )</td>
<td>2</td>
<td>6</td>
<td>( J_1 )</td>
</tr>
<tr>
<td>( \tau_2 )</td>
<td>4</td>
<td>9</td>
<td>( J_2 )</td>
</tr>
<tr>
<td>( J_3 )</td>
<td>15</td>
<td>1</td>
<td>( J_3 )</td>
</tr>
</tbody>
</table>

(a) 3 points compute maximum the server utilization \( U_S \) that can be used by a Total Bandwidth Server (TBS) without compromising the schedulability of the periodic tasks.

(b) 8 points compute the response times for the three jobs when being served by (plain) TBS.

(c) 8 points compute the response times for the three jobs when being served by optimized TBS.

(d) 3 points name the advantage(s) of a Constant Bandwidth Server over optimized TBS scheduling.

(e) 8 points compute the response times for the three jobs when being served by CBS.

Question 5 [10 points]

Considering the set of tasks below:

<table>
<thead>
<tr>
<th>( \tau )</th>
<th>( C_i )</th>
<th>( T_i^{min} )</th>
<th>( T_i^{max} )</th>
<th>( E_i )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \tau_1 )</td>
<td>5</td>
<td>8</td>
<td>15</td>
<td>1</td>
</tr>
<tr>
<td>( \tau_2 )</td>
<td>6</td>
<td>10</td>
<td>25</td>
<td>3</td>
</tr>
</tbody>
</table>

(a) 10 points Use the Elastic Period Method to derive the periods \( T_i \)'s that would guarantee the schedulability of these tasks under the Rate Monotonic Algorithm (LL bound). Assume that the initial period is \( T_i^{min} \) for both tasks.