First read the entire examination. There are 6 exercises in total. Grades are included between parentheses at all parts and sum up to 11 points. Good luck!

1. A recursive equation to determine the worst-case response time of a periodic task \( \tau_i \) is given by

\[
x = C_i + \sum_{j<i} \left\lfloor \frac{x}{T_j} \right\rfloor C_j.
\]

(a) (0.5) For which class of scheduling algorithms is this equation applicable?

**Answer** Fixed-priority pre-emptive scheduling (FPPS).

(b) (0.5) Give at least four assumptions that need to hold to use this equation.

**Answer** The equation is applicable for FPPS when deadlines are at most equal to periods, and task \( \tau_j \) has a higher priority than task \( \tau_i \) if and only if \( j < i \). See Section 4.1. of the book for general assumptions for periodic task scheduling.

(c) (1.0) Is the value \( \ell_i \) given by

\[
\ell_i = \frac{C_i}{1 - U_{i-1}}
\]

an appropriate initial value for the iterative procedure to determine the worst-case response time of \( \tau_i \)? Motivate your answer.

**Answer** Yes it is.

Let \( U_i = \sum_{j \leq i} \frac{C_i}{T_j} \). The worst-case response time \( WR_i \) is the smallest positive value satisfying the recursive equation, and the iterative procedure therefore starts with a lower bound. Hence, we have to prove that \( \ell_i \) is a lower bound for \( WR_i \), i.e. \( \ell_i \leq WR_i \).

To this end, we derive

\[
WR_i = C_i + \sum_{j<i} \left\lfloor \frac{WR_i}{T_j} \right\rfloor C_j \\
\geq C_i + \sum_{j<i} \frac{WR_i}{T_j} C_j \\
= C_i + WR_i \cdot \sum_{j<i} \frac{C_i}{T_j} = C_i + WR_i \cdot U_{i-1}.
\]

Hence, for \( U_{i-1} < 1 \), we get \( WR_i \geq \frac{C_i}{1 - U_{i-1}} \).

Note that because \( \ell_i \) is an appropriate initial value for \( U_i = \sum_{j \leq i} \frac{C_i}{T_j} \), we immediately see that it is also an appropriate value for the alternative interpretation \( U_i = \frac{C_i}{T_i} \), where \( U_0 = 0 \).
2. Consider (fixed-priority) servers.

(a) (0.5) Describe the purpose of a server.
Answer See book.

(b) (0.5) Describe how a server compares to background scheduling.
Answer See book.

(c) (0.5) Under which conditions does a polling server behave as a periodic task?
Answer When there is always work pending for the polling server if its capacity is larger than zero.

3. Consider a hybrid set of tasks $\Gamma$, consisting of a set $\Gamma_H$ of $n$ hard real-time periodic tasks and a set $\Gamma_S$ of soft real-time tasks. The hard-real time tasks are denoted by $\tau_1, \tau_2, \ldots, \tau_n$. We assume fixed-priority preemptive scheduling and arbitrary phasings of hard real-time tasks. A hard real-time task $\tau_i$ is characterized by a period $T_i$, a worst-case computation time $C_i$, and a deadline $D_i$. For notational convenience, we assume that task $\tau_j$ has a higher priority than task $\tau_i$ if and only if $j < i$.

Consider a system with a deferable server $DS$ for handling the soft real-time tasks next to the hard real-time tasks. The server has a capacity $C_{DS}$ and a period $T_{DS}$. Assume that the server receives the highest priority.

(a) (1.5) Derive an equation to determine the worst-case response times of the hard real-time tasks.

(b) Consider a soft real-time task $\tau_s$ with a computation time $C_s = 2C_{DS}$. Assume that the task can immediately start upon its release.

i. (0.5) Determine the longest response time of $\tau_s$.

ii. (0.5) Determine the shortest response time of $\tau_s$.

Answers This question has been taken from the examination of June 23rd, 2005.

4. Consider the following taskset.

<table>
<thead>
<tr>
<th>Name</th>
<th>Priority</th>
<th>Computation time</th>
<th>Resources</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\tau_1$</td>
<td>1</td>
<td>$C_1$</td>
<td>$Ra, Rb$</td>
</tr>
<tr>
<td>$\tau_2$</td>
<td>2</td>
<td>$C_2$</td>
<td>none</td>
</tr>
<tr>
<td>$\tau_3$</td>
<td>3</td>
<td>$C_3$</td>
<td>$Ra$</td>
</tr>
<tr>
<td>$\tau_4$</td>
<td>4</td>
<td>$C_4$</td>
<td>$Rb$</td>
</tr>
</tbody>
</table>

The priorities are fixed with lower numbers representing higher priorities. The tasks need resources that they reserve through a regular locking mechanism that is not of our concern here. For example, task $\tau_1$ needs resources $Ra$ and $Rb$ for its operations and acquires them in that order.

(a) (1.0) Discuss scenarios for priority inversion. Is bounded inversion possible?
(b) (1.0) Assume that we use the priority inheritance protocol to resolve this. What is the maximum blocking time for \( \tau_1 \) expressed in the given computation times? And what is this value if we use the priority ceiling protocol?

**Answers** This question has been taken from the examination of June 23\(^{rd} \), 2004.

5. The course explicitly distinguishes between tasks (related to the model) and processes and threads (platform related notions).

   (a) (0.5) Give three reasons for introducing tasks during design.

   (b) (0.5) Explain the differences between processes and threads by giving at least two characteristics of each.

**Answers** See slides ‘RTA.C7-mapping.pdf’.

6. One of the lectures concerned *A QoS approach for multimedia consumer terminals with media processing in software*.

   (a) (1.0) Explain which real-time problems were addressed.

   (b) (1.0) Explain how these problems have been solved.

**Answers** See slides of the lecture.