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>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Points:</td>
<td>10</td>
<td>15</td>
<td>20</td>
<td>10</td>
<td>5</td>
<td>15</td>
<td>5</td>
<td>5</td>
<td>85</td>
</tr>
<tr>
<td>Score:</td>
<td></td>
<td></td>
<td></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
• 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
### Liu and Layland (LL) test

\[ U \leq n(2^{1/n} - 1) \]

### Hyperbolic bound (HB) test

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

### Response-Time Analysis (FP)

#### No jitter

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

\[
R_i^{(0)} = C_i
\]

#### With jitter

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

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

\[ w^+ = \max(w, 0) \]

### Demand Bound Analysis

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

\[
g(0, L) = \sum_{i=1}^{n} \left\lfloor \frac{L + T_i - D_i}{T_i} \right\rfloor C_i
\]

### Schedulability

\[
\forall L \in \overline{D}, \quad g(0, L) \leq L
\]

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

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

\[ D_{max} = \max\{D_i | \tau_i \in \tau\} \]

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

#### CBS Server: Arrival of a job \( J_k \)

if (\( \exists \) a pending aperiodic job) then enqueue \( J_k \);
else if (\( q_s \geq (d_s - a_k) \cdot U_s \)) then
  \{ \( q_s \leftarrow Q_s; \quad d_s \leftarrow \max\{d_s, a_k\} + T_s; \) \}
\[ q_s \leftarrow Q_s \]
\[ d_s \leftarrow d_s + T_s \]

#### Budget exhaustion

### Response-time NP-FP

pre-requisites: \( D \leq T \) and preemptive-schedulable

\[
WO_i^{(n)} = B_i + \sum_{k=1}^{i-1} \left( \left\lfloor \frac{WO_i^{(n-1)}}{T_k} \right\rfloor + 1 \right) \cdot C_k
\]

\[ B_i = \max\{C_j | \forall \tau_j, P_i < P_j\} \]

\[ R_i = C_i + WO_i \]

\[
WO_i^{(0)} = B_i + \sum_{k=1}^{i-1} C_k
\]

### Response-time NP-EDF

\[
\forall \tau_i, 1 < i \leq n, \forall L, T_1 < L < T_i:
\]

\[
L \geq C_i + \sum_{k=1}^{i-1} \left\lfloor \frac{L - 1}{T_k} \right\rfloor \cdot C_k
\]
Question 1

Which task sets are schedulable by Rate Monotonic scheduling?

Solution:

(i) Feasible as HB = \((1 + 1/3) \times (1 + 1/4) \times (1 + 1/5) = 4/3 \times 5/4 \times 6/5 = 2\)

(ii) Feasible as HB will be lower than for (i), which was already fine.

(iii) Infeasible as follows from the critical instant

(iv) Feasible as follows from (iii) that showed a Lateness of 1 by task \(\tau_3\).

Which task sets are schedulable by Earliest Deadline First scheduling?

Solution: Tasks sets (i), (ii), and (iv) are feasible as RM can do them and EDF is optimal. Task set (iii) is also feasible as follows from the processor utilization \((1/3 + 1/4 + 2/5 = 59/60)\) being less than 1.

Question 2

Consider the following task set:

<table>
<thead>
<tr>
<th>(C_i)</th>
<th>(D_i)</th>
<th>(T_i)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\tau_1)</td>
<td>2</td>
<td>5</td>
</tr>
<tr>
<td>(\tau_2)</td>
<td>3</td>
<td>6</td>
</tr>
<tr>
<td>(\tau_3)</td>
<td>5</td>
<td>15</td>
</tr>
</tbody>
</table>

(a) Use the processor demand criterion to demonstrate that the above task set is unfeasible under EDF scheduling.
Solution:
Processor utilization is 1 (!), so $L^*$ can’t be computed and we need to go as far as the Hyperperiod (42).

<table>
<thead>
<tr>
<th>demand</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$g(0,5)$</td>
<td>2</td>
</tr>
<tr>
<td>$g(0,6)$</td>
<td>5</td>
</tr>
<tr>
<td>$g(0,11)$</td>
<td>7</td>
</tr>
<tr>
<td>$g(0,13)$</td>
<td>10</td>
</tr>
<tr>
<td>$g(0,15)$</td>
<td>15</td>
</tr>
<tr>
<td>$g(0,17)$</td>
<td>17</td>
</tr>
<tr>
<td>$g(0,20)$</td>
<td>20</td>
</tr>
<tr>
<td>$g(0,23)$</td>
<td>22</td>
</tr>
<tr>
<td>$g(0,27)$</td>
<td>25</td>
</tr>
<tr>
<td>$g(0,29)$</td>
<td>27</td>
</tr>
<tr>
<td>$g(0,34)$</td>
<td>30</td>
</tr>
<tr>
<td>$g(0,35)$</td>
<td>32</td>
</tr>
<tr>
<td>$g(0,36)$</td>
<td>37</td>
</tr>
</tbody>
</table>

Task $\tau_3$ (which must complete by $t = 36$) misses its deadline!

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

**Solution:**

<table>
<thead>
<tr>
<th>Task</th>
<th>Response time</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\tau_1$</td>
<td>5</td>
</tr>
<tr>
<td>$\tau_2$</td>
<td>7</td>
</tr>
<tr>
<td>$\tau_3$</td>
<td>16</td>
</tr>
</tbody>
</table>

Alternative solution is to schedule task $\tau_2$ at time $t = 36$ allowing it to finish within its deadline, but then task $\tau_1$ incurs a deadline violation.

<table>
<thead>
<tr>
<th>Task</th>
<th>Response time</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\tau_1$</td>
<td>6</td>
</tr>
<tr>
<td>$\tau_2$</td>
<td>6</td>
</tr>
<tr>
<td>$\tau_3$</td>
<td>16</td>
</tr>
</tbody>
</table>

**Question 3** [20 points]

Consider the following periodic tasks and aperiodic jobs:

<table>
<thead>
<tr>
<th></th>
<th>$C_i$</th>
<th>$T_i$</th>
<th></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>
<td>6</td>
<td>2</td>
</tr>
<tr>
<td>$\tau_2$</td>
<td>1</td>
<td>6</td>
<td>$J_2$</td>
<td>10</td>
<td>2</td>
</tr>
</tbody>
</table>
The system designer wants to use fixed-priority servers to schedule the aperiodic jobs. Assume the server is configured as \( \langle C_s = 2, T_s = 5 \rangle \) and the rate-monotonic scheduling is used to schedule the tasks and the server.

(a)  6 points  Can a **polling server** be used without compromising the feasibility of the periodic tasks? Determine the worst-case response times of the two jobs.

**Solution:**

The periodic tasks are fine if adding the polling server still results in a feasible RM schedule. Util = 0.9 ≤ 1, but HB = 4/3 * 7/6 * 7/5 = 196/90 > 2, so we have to resort to the critical instant to show feasibility.

![Timeline of polling server](image1)

The response times \( (J_1 = 6, J_2 = 7) \) follow from the timeline:

![Timeline of response times](image2)

(b)  6 points  Can a **deferable server** be used without compromising the feasibility of the periodic tasks? Determine the worst-case response times of the two jobs.

**Solution:** Compared to a polling server we now face the issue of jitter as the deferable server schedules a task as soon as it gets the chance. We can compute the worst case response time of task \( \tau_2 \) factoring in a jitter of 5 – 2 = 3 for the server iterating towards a fixed point of 8: \( R_2 = 1 + \left\lceil \frac{8+3}{5} \right\rceil + \left\lceil \frac{8}{5} \right\rceil 2 = 1 + 3 + 4 = 8 \). Thus, the configuration with a polling server may (depending on the actual job arrivals) compromise the timely execution of task \( \tau_2 \). (The worst case response time of task \( \tau_1 \) can be iterative computed as \( R_1 = 2 + \left\lceil \frac{4+3}{5} \right\rceil = 4 \).

The response times \( (J_1 = 2, J_2 = 2) \) follow from the timeline (notice the deadline miss of task \( \tau_2 \):
(c) **4 points** Now consider a fixed-priority scheduling. What is the highest possible priority that one can assign to this **deferable server** such that none of the periodic tasks miss their deadline in the example above? (your answer must include the priority of each of the tasks as well as the server.)

(d) **4 points** What is a disadvantage of a **polling server** and how a **deferable server** can improve it?

**Solution:** If aperiodic tasks come later than the release time of the server, they have to wait for the next release of the server! Deferable server keeps the budget even if there is no aperiodic task in the queue.

**Question 4** [10 points]

To schedule the following aperiodic jobs ($J_1$ and $J_2$), a constant bandwidth server with the period $T_s = 6$ and budget $Q_s = 2$ is used.

<table>
<thead>
<tr>
<th>a_i</th>
<th>$WCET_i$</th>
<th>$T_i$</th>
<th>type</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\tau_1$</td>
<td>0</td>
<td>2</td>
<td>6</td>
</tr>
<tr>
<td>$\tau_2$</td>
<td>0</td>
<td>3</td>
<td>9</td>
</tr>
<tr>
<td>$J_1$</td>
<td>2</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>$J_2$</td>
<td>2</td>
<td>12</td>
<td></td>
</tr>
</tbody>
</table>

where $a_i$ denotes the arrival time of the job (or the phase of the periodic task). Note that for periodic tasks, the deadline is equal to the period.
(a) 10 points Draw the schedule from time 0 to time 18.

**Solution:**

![Schedule Diagram]

**Question 5**

Consider a system scheduled by a fixed-priority scheduling in which there is a polling server that has the highest priority among all periodic tasks.

(a) 5 points Generate an example of a periodic task set and an aperiodic job for which the response time of the aperiodic job is larger when it is scheduled within the polling server than when it is scheduled as a background service. Draw the schedule of your task set together with the job to justify your answer.

**Solution:** The polling server reduces its budget to zero if there is no priority task in the system. See the example:

![Polling Server Diagram]

**Question 6**

(a) 2 points What is the advantage of fixed-priority scheduling over EDF when there might be overloads in the system?

**Solution:** see the slides.

(b) 3 points Briefly explain what is the cache-related preemption delay and how does it impact the schedulability.
(c) **3 points** What is the length of critical path in the following directed-acyclic graph (DAG)? Assume that the values written on the vertices are the WCET of the code segment and assume there is an infinite number of processors.

![Directed-acyclic graph](image)

**Solution:** it is $1+2+8+3 = 14$

(d) **3 points** Consider the highest-locker priority (HLP) protocol for real-time tasks that use shared resources protected by semaphores. List one advantage of HLP over the Non-preemptive Resource Access Protocol (NPP) and one disadvantage of HLP (in general).

**Solution:** see the slides.

(e) **4 points** List two advantage of partitioned multiprocessor scheduling over global multiprocessor scheduling.

**Solution:** see the slides.

---

**Question 7**

5 points

Answer the following multiple-choice questions. Note that some questions have fewer than 4 choices.

(a) **1 point**

**Claim:** "A **deferable server** may impact the schedule of the higher-priority periodic tasks"

**Cause:** "because it keeps its budget even if there is no aperiodic jobs and hence it may schedule two aperiodic jobs consecutively (one at the end of the server’s period and one at the beginning of the next activation of the server). It results in a long delay for the periodic tasks."

(a) claim is true, cause is true  (b) claim is true, cause is false  (c) claim is false

**Solution:** answer: c

(b) **1 point**

**Claim:** A periodic task set with $U = 1$ is never schedulable by the preemptive rate-monotonic scheduling algorithm.

**Cause:** because the Liu and Layland test will certainly reject that task set.

(a) claim is true, cause is true  (b) claim is true, cause is false  (c) claim is false

**Solution:** answer: c
In the following demand-bound function (DBF) that has been drawn for a periodic task set with two tasks, what are the WCET and deadline of the task whose period is 4? (hint: the other task has a period larger than 4).

(a) \( C_1 = 1, D_1 = 3 \)  
(b) \( C_1 = 2, D_1 = 3 \)  
(c) \( C_1 = 2, D_1 = 4 \)  
(d) \( C_1 = 1, D_1 = 4 \)

\[ \tau_1: (C_1 = ?, T_1 = 4, D_1 = ?) \]

This is the demand bound function for two tasks.

Tasks are indexed by period so \( T_1 < T_2 \).

Solution: answer: a

Claim 1: If a task set does not pass the hyperbolic-bound test, then it will also not pass the Liu and Layland test.

Claim 2: If a task set is harmonic and feasible (i.e., \( U < 1 \)), then the hyperbolic-bound test will certainly accept it.

(a) claim 1 is true, claim 2 is true  
(b) claim 1 is true, claim 2 is false  
(c) claim 1 is false, claim 2 is true  
(d) claim 1 is false, claim 2 is false

Solution:
answer: b because the hyperbolic bound and LL tests ignore period relations. To them, only the utilization matters. Both of them may reject the task set.

Assume that the following task set is scheduled using global non-preemptive fixed-priority algorithm with priorities \( P_2 < P_1 < P_3 < P_4 \) on a multicore platform with 3 cores. What is the worst-case response time of \( \tau_1 \)?

(a) 1  
(b) 2  
(c) 3  
(d) 4

Solution:
answer: d
Question 8 [5 points]

The following question is optional and acts as a bonus question!

(a) [5 points] Briefly explain how online preemptive fixed-priority scheduling can be implemented in a constant runtime regardless of the number of tasks and priorities in the system.

Solution:

- However,

  Using a **bitmap data structure** + **most-significant-bit instruction**

  ⇒ **FP can be implemented in O(1)**

  (provided that the number of priority levels is limited by a constant)

---

**Implementing FP using bitmaps:**

1. Whenever a task is **released**, set its bit in the flag to 1
2. Whenever a task **completes**, set its bit in the flag to 0
3. To **find** the highest-priority ready task, calculate the **most-significant-bit** of the flag

---

![Diagram of online scheduler with ready queue and tasks](image)