Embedded Systems

Exercise 3:
Scheduling Real-Time Periodic and Mixed Task Sets

13/20. April 2016

Pengcheng Huang
pengcheng.huang@tik.ee.ethz.ch
Overview

• Basic definitions in Real-Time (2 slides)

• Scheduling periodic tasks
  - Rate Monotonic (3 slides)
  - Deadline Monotonic (4 slides)
  - Earliest Deadline First (2 slides)

• Mixed tasks
  - Polling Server (2 slides)
  - Total Bandwidth Server (2 slides)
Definitions: RT Periodic Tasks

Arrival Time

Deadline, \( D_i \)

Execution Time, \( C_i \)

Period, \( T_i \)
Some Assumptions

Tasks are independent

All instances of a periodic task have

- same worst-case execution time, $C_i$
- same relative deadline $D_i \leq T_i$ (often $D_i = T_i$)

... see lecture notes for the full list of assumptions
# Scheduling Problems and Algorithms

<table>
<thead>
<tr>
<th>Static Priority</th>
<th>Periodic with $D = T$</th>
<th>Periodic with $D &lt; T$</th>
<th>Mixed tasks</th>
</tr>
</thead>
<tbody>
<tr>
<td>RM</td>
<td>DM</td>
<td>Polling Server</td>
<td></td>
</tr>
<tr>
<td>Dynamic Priority</td>
<td>EDF</td>
<td>EDF</td>
<td>Total Bandwidth Server</td>
</tr>
</tbody>
</table>

**Note:** The tasks can be periodic or sporadic, and the algorithms can be static or dynamic.
# Scheduling Problems and Algorithms

<table>
<thead>
<tr>
<th>Static Priority</th>
<th>Periodic with $D = T$</th>
<th>Periodic with $D &lt; T$</th>
<th>Mixed tasks</th>
</tr>
</thead>
<tbody>
<tr>
<td>RM</td>
<td>DM</td>
<td>Polling Server</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Dynamic Priority</th>
<th>Total Bandwidth Server</th>
</tr>
</thead>
<tbody>
<tr>
<td>EDF</td>
<td></td>
</tr>
</tbody>
</table>
Rate Monotonic (RM)

Static priority assignment rule:
Tasks with shorter period get higher priority
RM - Schedulability Analysis

\[ \sum_{i=1}^{n} \frac{C_i}{T_i} \leq n \left(2^{\frac{1}{n}} - 1\right) \]

sufficient (but not necessary)
Necessary vs Sufficient

• $X \Rightarrow Y \iff X$ is sufficient for $Y$
• $\neg (X) \Rightarrow \neg (Y) \iff X$ is necessary for $Y$

Test:

• Is sunlight a necessary or sufficient condition for the roses to bloom?
• Is having the flu virus in your blood a necessary or sufficient condition for being sick?
• Is attending the exercise a necessary or sufficient condition for passing the course?
## Scheduling Problems and Algorithms

<table>
<thead>
<tr>
<th></th>
<th>Periodic with $D = T$</th>
<th>Periodic with $D &lt; T$</th>
<th>Mixed tasks</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Static Priority</strong></td>
<td>RM</td>
<td>DM</td>
<td>Polling Server</td>
</tr>
<tr>
<td><strong>Dynamic Priority</strong></td>
<td>EDF</td>
<td>EDF*</td>
<td>Total Bandwidth Server</td>
</tr>
</tbody>
</table>

**Periodic with $D = T$**

**Periodic with $D < T$**

**Mixed tasks**

**Polling Server**

**Total Bandwidth Server**
Deadline Monotonic (DM)

Static priority assignment rule:
Tasks with smaller relative deadline get higher priority

T2

T3

T1
Deadline Monotonic (DM)

Static priority assignment rule:
Tasks with smaller relative deadline get higher priority

High

T2

T3

T1

Low
DM - Schedulability Analysis (1)

\[ \sum_{i=1}^{n} \frac{C_i}{D_i} \leq n \left( 2^{\frac{1}{n}} - 1 \right) \]

sufficient 
(but not necessary)
DM - Schedulability Analysis (2)

Worst case interference:

\[ I_i = \sum_{j=1}^{i-1} \left\lfloor \frac{R_i}{T_j} \right\rfloor C_j + C_i \]

Algorithm: DM\_guarantee (\( \Gamma \))

\{
  \text{for (each } \tau_i \in \Gamma)\{
    I = 0;
    \text{do } \{
      R = I + C_i;
      \text{if (} R > D_i \text{) return (UNSCHEDULABLE);}\
      I = \sum_{j=1, \ldots, (i-1)} \left\lfloor \frac{R}{T_j} \right\rfloor C_j;
      \text{while (} I + C_i > R)\
    \text{\}}
  \text{\}}
  \text{return (SCHEDULABLE);}\
\}

necessary & sufficient
Scheduling Problems and Algorithms

Periodic with $D = T$

Periodic with $D < T$

Mixed tasks

Static Priority
- RM
- DM
- Polling Server

Dynamic Priority
- EDF
- EDF
- Total Bandwidth Server
Earliest Deadline First (EDF)

Always execute the task with the currently closest deadline
EDF - Schedulability Analysis

\[ D_i = T_i \]

\[ D_i < T_i \]

necessary & sufficient

\[ \sum_{i=1}^{n} \frac{C_i}{T_i} \leq 1 \]

sufficient (but not necessary)

\[ \sum_{i=1}^{n} \frac{C_i}{D_i} \leq 1 \]

Demand bound analysis is a necessary analysis.
Reference to paper in solution
# Scheduling Problems and Algorithms

<table>
<thead>
<tr>
<th></th>
<th>Periodic with $D = T$</th>
<th>Periodic with $D &lt; T$</th>
<th>Mixed tasks</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Static Priority</strong></td>
<td>RM</td>
<td>DM</td>
<td>Polling Server</td>
</tr>
<tr>
<td><strong>Dynamic Priority</strong></td>
<td>EDF</td>
<td>EDF</td>
<td>Total Bandwidth Server</td>
</tr>
</tbody>
</table>
Mixed Task Scheduling

- Mixed (or hybrid) task-set has both periodic and aperiodic tasks

- Basic idea to schedule them:
  - Serve the periodic tasks as usual
  - Serve the aperiodic tasks via a “server” that behaves like a periodic task but serves the aperiodic task

- We see two examples of servers. Several others exist.
## Scheduling Problems and Algorithms

<table>
<thead>
<tr>
<th></th>
<th>Periodic with $D = T$</th>
<th>Periodic with $D &lt; T$</th>
<th>Mixed tasks</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Static Priority</strong></td>
<td>RM</td>
<td>DM</td>
<td>Polling Server</td>
</tr>
<tr>
<td><strong>Dynamic Priority</strong></td>
<td>EDF</td>
<td>EDF</td>
<td>Total Bandwidth Server</td>
</tr>
</tbody>
</table>
RM - Polling Server

Basic idea:
Introduce an artificial periodic task — the server for aperiodic requests

<table>
<thead>
<tr>
<th>$\tau_1$</th>
<th>$C_i$</th>
<th>$T_i$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\tau_1$</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>$\tau_2$</td>
<td>2</td>
<td>6</td>
</tr>
</tbody>
</table>

Server

$C_s = 2$
$T_s = 5$
RM - Polling Server

- Schedulability analysis of periodic tasks:

\[
\frac{C_S}{T_S} + \sum_{i=1}^{n} \frac{C_i}{T_i} \leq (n + 1) \left(2^{\frac{1}{n+1}} - 1\right)
\]

- Guarantee of aperiodic activities:

\[
(1 + \left\lceil \frac{C_a}{C_S} \right\rceil)T_S \leq D_a
\]

sufficient
(but not necessary)
## Scheduling Problems and Algorithms

<table>
<thead>
<tr>
<th>Periodic with $D = T$</th>
<th>Periodic with $D &lt; T$</th>
<th>Mixed tasks</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Static Priority</strong></td>
<td><strong>Dynamic Priority</strong></td>
<td>Total Bandwidth Server</td>
</tr>
<tr>
<td>RM</td>
<td>EDF</td>
<td></td>
</tr>
<tr>
<td>DM</td>
<td>EDF</td>
<td></td>
</tr>
<tr>
<td>Polling Server</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Basic idea: Dynamically assign deadlines to aperiodic tasks such that the server utilization factor $U_s$ (bandwidth) is never exceeded

**Example:** $U_p = 0.75$, $U_s = 0.25$, $U_p + U_s = 1$
k-th aperiodic request receives deadline

\[ d_k = \max(r_k, d_{k-1}) + \frac{C_k}{U_s} \]
Problem 1

Aufgabe 1: Schedulability Test for Earliest Deadline First (EDF)

<table>
<thead>
<tr>
<th></th>
<th>( \tau_1 )</th>
<th>( \tau_2 )</th>
<th>( \tau_3 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( C_i )</td>
<td>6</td>
<td>5</td>
<td>1</td>
</tr>
<tr>
<td>( T_i )</td>
<td>9</td>
<td>15</td>
<td>5</td>
</tr>
</tbody>
</table>

1. Test if the given task-set is schedulable with EDF. Is this test necessary and sufficient?
2. Assume that the first job of each task arrives at time 0. Construct the schedule for the interval \([0, 20]\) and illustrate it graphically. In case they exist, identify deadline misses.
3. If there are deadline misses for some task(s) in the constructed schedule, then will deadline misses always be confined to the same task(s)?
Solution 1

<table>
<thead>
<tr>
<th></th>
<th>$T_1$</th>
<th>$T_2$</th>
<th>$T_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C_i$</td>
<td>6</td>
<td>5</td>
<td>1</td>
</tr>
<tr>
<td>$T_i$</td>
<td>9</td>
<td>15</td>
<td>5</td>
</tr>
</tbody>
</table>

\[ \sum_{i=1}^{n} \frac{C_i}{T_i} \leq 1 \]

\[
\frac{6}{9} + \frac{5}{15} + \frac{1}{5} = 1.2 \not\leq 1 \text{ failed!}
\]
Solution 1

No result on which task will miss deadline!
Problem 2

Aufgabe 2: Schedulability Test for Fixed Priorities - Rate Monotonic (RM)

<table>
<thead>
<tr>
<th></th>
<th>$\tau_1$</th>
<th>$\tau_2$</th>
<th>$\tau_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C_i$</td>
<td>1</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>$T_i$</td>
<td>3</td>
<td>8</td>
<td>9</td>
</tr>
</tbody>
</table>

1. Test if the given task-set is schedulable under RM, using the sufficient test.
2. Test if the given task-set is schedulable under RM, using the necessary test.
3. Assume that the first job of each task arrives at time 0. Construct the schedule for the interval $[0, 20]$ and illustrate it graphically. In case they exist, identify deadline misses.
Solution 2

<table>
<thead>
<tr>
<th></th>
<th>T₁</th>
<th>T₂</th>
<th>T₃</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C_i$</td>
<td>1</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>$T_i$</td>
<td>3</td>
<td>8</td>
<td>9</td>
</tr>
</tbody>
</table>

\[
\sum_{i=1}^{n} \frac{C_i}{T_i} \leq n \left(2^{\frac{1}{n}} - 1\right)
\]

\[
1/3 + 3/8 + 2/9 = 0.93 \neq 3(2^{1/3} - 1) = 0.78 \text{ failed!}
\]
Solution 2

\[ T_3: \]
\[
\begin{align*}
R_3^0 &= C_3 = 2 & l_3^0 &= \left[ \frac{2}{3} \right] 1 + \left[ \frac{2}{3} \right] 3 = 1 + 3 = 4 & 4 + 2 \neq 2 \\
R_3^1 &= 4 + 2 = 6 & l_3^1 &= \left[ \frac{4}{3} \right] 1 + \left[ \frac{4}{3} \right] 3 = 2 + 3 = 5 & 5 + 2 \neq 6 \\
R_3^2 &= 5 + 2 = 7 & l_3^2 &= \left[ \frac{5}{3} \right] 1 + \left[ \frac{5}{3} \right] 3 = 3 + 3 = 6 & 6 + 2 \neq 7 \\
R_3^3 &= 6 + 2 = 8 & l_3^3 &= \left[ \frac{6}{3} \right] 1 + \left[ \frac{6}{3} \right] 3 = 3 + 3 = 6 & 6 + 2 = 8 \ldots \text{OK (since } 8 \leq T_3 = 9) \\
\end{align*}
\]

\[ T_2: \]
\[
\begin{align*}
R_2^0 &= C_2 = 3 & l_2^0 &= \left[ \frac{3}{3} \right] 1 = 1 & 1 + 3 \neq 3 \\
R_2^1 &= 1 + 3 = 4 & l_2^1 &= \left[ \frac{4}{3} \right] 1 = 2 & 2 + 3 \neq 4 \\
R_2^2 &= 2 + 3 = 5 & l_2^2 &= \left[ \frac{5}{3} \right] 1 = 2 & 2 + 3 = 5 \ldots \text{OK (since } 5 \leq T_2 = 8) \\
\end{align*}
\]

\[ T_1: \]
\[
\begin{align*}
R_1^0 &= C_1 = 1 & l_1^0 &= 0 & \ldots \text{OK}
\end{align*}
\]
Solution 2
Problem 3 (part 1: analysis)

Aufgabe 3: Scheduling with Polling Server

<table>
<thead>
<tr>
<th></th>
<th>$\tau_1$</th>
<th>$\tau_2$</th>
<th>$\tau_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C_i$</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>$D_i$</td>
<td>6</td>
<td>8</td>
<td>16</td>
</tr>
<tr>
<td>$T_i$</td>
<td>6</td>
<td>8</td>
<td>16</td>
</tr>
</tbody>
</table>

In addition to the above periodic tasks, we have an aperiodic job $J_a$ with computation time $C_a = 1$, and relative deadline $D_a$. The scheduling policy is RM. The aperiodic job is scheduled through a Polling Server (PS).

1. Let the period and computing time of the polling server be 25 and 1, respectively. Compute the aperiodic guarantee available to $J_a$, i.e., compute the relative deadline of $J_a$ which is guaranteed not to be missed.
2. Using the sufficient test of RM, test if the polling server of 1. is schedulable along with the periodic task-set?
3. Determine integral parameters of the polling server such that (a) the relative deadline guaranteed to $J_a$ is minimised, and (b) the RM schedule satisfies the sufficient schedulability test.
4. For the optimal setting of 3. devise a necessary schedulability test with the relative deadline of the aperiodic task $D_a = 32$. 
Solution 3 (part 1 : analysis)

- Given polling server parameters, compute aperiodic guarantee

\[
(1 + \left[ \frac{C_a}{C_s} \right]) T_s \leq D_a
\]

\[
D_a = 50
\]
Solution 3 (part 2: design)

\[
\frac{C_s}{T_s} + \sum_{i=1}^{n} \frac{C_i}{T_i} \leq (n + 1)(2^{\frac{n+1}{n+1}} - 1)
\]

\[
\frac{C_s}{T_s} \leq 0.76 - 0.708 = 0.048
\]

Can choose \((C_s, T_s) = (1, 21)\) or \((2, 42)\) or \((3, 63)\)

Smallest delay \(D_a\) for \((1, 21)\) setting

\[
(1 + \left\lceil \frac{C_a}{C_s} \right\rceil)T_s \leq D_a
\]
### Solution 3 (part 2: design)

**Can do tighter analysis**

<table>
<thead>
<tr>
<th>$R_a^0 = C_a = 1$</th>
<th>$I_a^0 = \left[\frac{1}{6}\right]2 + \left[\frac{1}{8}\right]2 + \left[\frac{1}{16}\right]2 = 6$</th>
<th>$6 + 1 \neq 1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$R_a^1 = 7$</td>
<td>$I_a^1 = \left[\frac{7}{6}\right]2 + \left[\frac{7}{8}\right]2 + \left[\frac{7}{16}\right]2 = 8$</td>
<td>$8 + 1 \neq 7$</td>
</tr>
<tr>
<td>$R_a^2 = 9$</td>
<td>$I_a^2 = \left[\frac{9}{6}\right]2 + \left[\frac{9}{8}\right]2 + \left[\frac{9}{16}\right]2 = 10$</td>
<td>$10 + 1 \neq 9$</td>
</tr>
<tr>
<td>$R_a^2 = 11$</td>
<td>$I_a^3 = \left[\frac{11}{6}\right]2 + \left[\frac{11}{8}\right]2 + \left[\frac{11}{16}\right]2 = 10$</td>
<td>$10 + 1 = 11$</td>
</tr>
</tbody>
</table>

The task can come just after the start of the period

$$D_a = 11 + T_s = 11 + 21 = 32$$