Embedded Systems SS 2003

Exercise 12: Periodic Scheduling, Mixed Task Sets

Issue Date: 18. June 2003
Discussion Date: 25. June 2003

Task 1: Periodic Scheduling with EDF

Given are three periodic tasks, with execution times and periods according to following table (the relative deadlines correspond to the periods).

<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>4</td>
<td>5</td>
<td>2</td>
</tr>
<tr>
<td>$T_i$</td>
<td>9</td>
<td>16</td>
<td>5</td>
</tr>
</tbody>
</table>

1. Verify if the given set task is schedulable
2. Construct the schedule, and illustrate it graphically. In case they exist, identify overflow situations.
Solution - Task 1

EDF - Earliest Deadline First scheduling algorithm is preemptive, and works under assumption that $D_i \leq T_i$. The currently executing task is preempted whenever another aperiodic instance with earlier deadline becomes active.

EDF is an optimal scheduling algorithm: no other algorithm can schedule a set of periodic tasks if the set cannot be scheduled by EDF.

The schedule is feasible if the sufficient and necessary condition is satisfied, namely if and only if

$$U = \sum_{i=1}^{n} \frac{C_i}{T_i} \leq 1$$

The term $U$ is the average processor utilization. In our case: $\frac{4}{9} + \frac{5}{16} + \frac{2}{5} = 1.116 \not\leq 1$

There is no valid schedule. The schedule (and the overflow situation) is presented in the Figure 1.

Figure 1: EDF schedule (not feasible).
Task 2: Periodic Scheduling with Fixed Priorities - RM

Given is the set of three periodic tasks (their relative deadlines equal their periods). Assign priorities to the tasks according to the Rate Monotonic (RM) scheduling scheme.

<table>
<thead>
<tr>
<th></th>
<th>τ₁</th>
<th>τ₂</th>
<th>τ₃</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cₙ</td>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>Tₙ</td>
<td>4</td>
<td>6</td>
<td>10</td>
</tr>
</tbody>
</table>

1. Verify the schedulability (under RM) of the given task set, using one processor.
2. Construct the schedule graphically, and discuss the overflow situations, in case they exist.
Solution - Task 2

The priorities to tasks are assigned statically, before the actual execution of the task set. Rate Monotonic scheduling scheme assigns higher priority to tasks with smaller periods. It is preemptive (tasks are preempted by the higher priority tasks). It is an optimal scheduling algorithm; if a task set cannot be scheduled with RM, it cannot be scheduled by any fixed-priority algorithm.

The sufficient schedulability test is given by:

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

The term \( U \) is said to be the processor utilization factor (the fraction of the processor time spent on executing task set).

In our case: 
\[
\frac{1}{4} + \frac{2}{6} + \frac{3}{10} = 0.88 \not\leq 3 \left( 2^{1/3} - 1 \right) = 0.78!
\]

The above condition is not necessary; we can do a somewhat more involved sufficient and necessary condition test, as follows.

We have to guarantee that all the tasks can be scheduled, in any possible instance. In particular, if a task can be scheduled in its critical instances, then the schedulability guarantee condition holds (a critical instance of a task occurs whenever the task is released simultaneously with all higher priority tasks). In the following, we perform the schedulability analysis for each of tasks (for the details of the algorithm, please have a look at the page 99, Buttazzo’s book).

The tasks are first ordered by their priorities: \( \tau_1, \tau_2 \) and \( \tau_3 \).

\( \tau_3 \):
\[
\begin{align*}
R^0_3 &= C_3 = 3 \quad I^0_3 = \left\lceil \frac{3}{4} \right\rceil 1 + \left\lceil \frac{3}{6} \right\rceil 2 = 1 + 2 = 3 \quad 3 + 3 \neq 3 \\
R^1_3 &= 3 + 3 = 6 \quad I^1_3 = \left\lceil \frac{6}{4} \right\rceil 1 + \left\lceil \frac{6}{6} \right\rceil 2 = 2 + 2 = 4 \quad 4 + 3 \neq 6 \\
R^2_3 &= 4 + 3 = 7 \quad I^2_3 = \left\lceil \frac{7}{4} \right\rceil 1 + \left\lceil \frac{7}{6} \right\rceil 2 = 3 + 4 = 7 \quad 7 + 3 \neq 9 \\
R^3_3 &= 6 + 3 = 9 \quad I^3_3 = \left\lceil \frac{9}{4} \right\rceil 1 + \left\lceil \frac{9}{6} \right\rceil 2 = 3 + 4 = 7 \quad 7 + 3 = 10 \ldots OK (since 10 \leq T_3 = 10) \\
R^4_3 &= 7 + 3 = 10 \quad I^4_3 = \left\lceil \frac{10}{4} \right\rceil 1 + \left\lceil \frac{10}{6} \right\rceil 2 = 3 + 4 = 7 \quad 7 + 3 = 10 \ldots OK (since 10 \leq T_3 = 10)
\end{align*}
\]

\( \tau_2 \):
\[
\begin{align*}
R^0_2 &= C_2 = 2 \quad I^0_2 = \left\lceil \frac{2}{3} \right\rceil 1 = 1 \quad 1 + 2 \neq 2 \\
R^1_2 &= 1 + 2 = 3 \quad I^1_2 = \left\lceil \frac{3}{3} \right\rceil 1 = 1 \quad 1 + 2 = 3 \ldots OK (since 3 \leq T_2 = 6)
\end{align*}
\]

\( \tau_1 \):
\[
\begin{align*}
R^0_1 &= C_1 = 1 \quad I^0_1 = 0 \ldots OK
\end{align*}
\]

The schedule is represented graphically in the Figure 2.

![Figure 2: RM schedule.](image-url)
Task 3: Periodic Scheduling with Fixed Priorities - DM

Given the following set of periodic tasks:

<table>
<thead>
<tr>
<th></th>
<th>τ₁</th>
<th>τ₂</th>
<th>τ₃</th>
</tr>
</thead>
<tbody>
<tr>
<td>(C_i)</td>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>(D_i)</td>
<td>5</td>
<td>4</td>
<td>8</td>
</tr>
<tr>
<td>(T_i)</td>
<td>5</td>
<td>6</td>
<td>10</td>
</tr>
</tbody>
</table>

Verify the schedulability of the task set using the Deadline Monotonic (DM) algorithm and then construct the schedule graphically.
Solution - task 3

DM (Deadline Monotonic) scheduling scheme assigns priorities based on the relative deadlines of the periodic tasks. Higher priority corresponds to a task having an earlier deadline.

One first schedulability test is (sufficient, not necessary): $1/5 + 2/4 + 3/8 = 1.075 \not\leq 3(2^{1/3} - 1) = 0.78$!

The other test to do follows the lines of the solution of the previous exercise: the tasks are ordered with respect to their priorities $\tau_2, \tau_1, \tau_3$.

\[
\tau_3: \\
R_3^0 = C_3 = 3 \quad I_3^0 = \lceil \frac{3}{5} \rceil 1 + \lceil \frac{3}{6} \rceil 2 = 1 + 2 = 3 \quad 3 + 3 \neq 3 \\
R_3^1 = 3 + 3 = 6 \quad I_3^1 = \lceil \frac{6}{5} \rceil 1 + \lceil \frac{6}{6} \rceil 2 = 2 + 2 = 4 \quad 4 + 3 \neq 6 \\
R_3^2 = 4 + 3 = 7 \quad I_3^2 = \lceil \frac{7}{5} \rceil 1 + \lceil \frac{7}{6} \rceil 2 = 2 + 4 = 6 \quad 6 + 3 \neq 7 \\
R_3^3 = 6 + 3 = 9 \quad I_3^3 = \lceil \frac{9}{5} \rceil 1 + \lceil \frac{9}{6} \rceil 2 = 2 + 4 = 6 \quad 6 + 3 = 9 \ldots 9 \leq D_3 = 8!
\]

The schedule is not feasible. The graphical representation of the failed scheduling is presented below:

Figure 3: DM schedule (not feasible).
Task 4: Mixed Tasks - Polling Server

Two periodic tasks are given, with execution times and periods given in the following table (deadlines equal periods). The given set of tasks should be scheduled with the Rate Monotonic scheduling scheme.

<table>
<thead>
<tr>
<th></th>
<th>( \tau_1 )</th>
<th>( \tau_2 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( C_1 )</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>( T_1 )</td>
<td>5</td>
<td>8</td>
</tr>
</tbody>
</table>

Construct a schedule graphically for following aperiodic requests (a Polling Server should be introduced). The CPU utilization has to be maximized.

<table>
<thead>
<tr>
<th></th>
<th>( J_1 )</th>
<th>( J_2 )</th>
<th>( J_3 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( a_i )</td>
<td>2</td>
<td>7</td>
<td>9</td>
</tr>
<tr>
<td>( C_i )</td>
<td>3</td>
<td>2</td>
<td>1</td>
</tr>
</tbody>
</table>

Solution - task 4

For the task set to be schedulable with RM (and the introduction of the Polling Server), one should have the following condition satisfied:

\[
\frac{C_s}{T_s} + \left( \frac{1}{5} + \frac{2}{8} \right) \leq 3\left(2^{1/3} - 1\right)
\]

\[
\frac{C_s}{T_s} \leq 0.329
\]

With parameters: \( C_s = 1 \) and \( T_s = 4 \), the following schedule is feasible:

Figure 4: RM schedule - Polling Server.
Task 5: Mixed Tasks - Total Bandwidth Server

We have to design a system that schedules periodic tasks with EDF and employs a total bandwidth server to serve aperiodic requests. We know of one sporadic aperiodic request with computation time $C_a = 3$ and a relative deadline $D_a = 8$. What is the maximum processor utilization available for periodic tasks if we want to guarantee that this aperiodic task completes within its deadline.

Solution - task 5

For a schedule to be feasible, the sufficient and necessary condition to hold is $U_p + U_s \leq 1$. Therefore we have

$$U_p \leq 1 - U_s = 1 - \frac{C_a}{D_a} = 1 - \frac{3}{8} = \frac{5}{8}$$

The maximum processor utilization available for periodic tasks is $5/8 = 0.625$. 
Task 6: Periodic Scheduling

A processor is supposed to execute the following set of tasks described by their execution times $C_i$, relative deadlines $D_i$ and periods $T_i$:

<table>
<thead>
<tr>
<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>
</tr>
<tr>
<td>$D_i$</td>
<td>5</td>
<td>4</td>
</tr>
<tr>
<td>$T_i$</td>
<td>6</td>
<td>8</td>
</tr>
</tbody>
</table>

The scheduling algorithm to be adopted is any fixed-priority scheduling scheme.

1. Execute the sufficient schedulability test and calculate the result. What statement regarding schedulability can be made based on your result?
2. Execute the necessary and sufficient schedulability test and calculate the result. What statement regarding schedulability can be made based on your result?
3. If there is a feasible schedule for a given task set, construct it graphically.
Solution - task 6

(1)
A sufficient schedulability test for a given task set (and with respect to the fixed-priorities scheduling schemes, such as RM or DM) is given by:

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

In our case, we have: \( \frac{2}{5} + \frac{2}{4} + \frac{4}{8} = 1.2 \not\leq 0.78 = 3\left(2^{1/3} - 1\right) \).

(2)
A sufficient and necessary test for RM (or DM) is done by estimating the value of the longest response time \( R_i = C_i + I_i \) of a periodic task \( \tau_i \) at its critical instances.

(RM) The tasks are ordered with respect to their priorities (smaller period - higher priority): \( \tau_1, \tau_2, \tau_3 \).

The analysis for the task \( \tau_3 \):

\[ R_3^0 = C_3 = 4 \quad I_3^0 = \left\lceil \frac{4}{2}\right\rceil + \left\lfloor \frac{4}{2}\right\rfloor = 2 + 2 = 4 \quad \text{not schedulable with RM} \]

\[ R_3^1 = 4 + 4 = 8 \quad I_3^1 = \left\lceil \frac{8}{2}\right\rceil + \left\lfloor \frac{8}{2}\right\rfloor = 4 + 2 = 6 \quad \text{not schedulable with RM} \]

Since \( R_3^2 = 4 + 6 = 10 \not\leq D_3 = 8 \), the tasks are not schedulable with RM.

The task set is not schedulable either by the DM (the test is the same, and the ordering of the tasks with respect to their priorities, still maintains the task \( \tau_3 \) as the lowest priority task).
Task 7: Mixed Task Sets

A processor is supposed to execute the following set of tasks described by their execution times $C$, relative deadlines $D$ and periods $T$:

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

A Polling Server is added that allows the scheduling of additional firm aperiodic activities. Use the sufficient schedulability test for analyzing the behavior in case of a single aperiodic request $J_a$ with computation time $C_a = 1$ and relative deadline $D_a$.

1. Determine the parameters of the Polling Server such that the smallest deadline $D_a$ of $J_a$ can be guaranteed. Only integer values of the parameters are allowed.
2. Determine parameters of the Polling Server such that only a higher deadline than that determined in 1. can be guaranteed.

Solution - task 7

Sufficient schedulability test for RM (and a Polling Server introduced to handle the firm aperiodic request $J_a$) is given by:

$$\sum_{i=1}^{n} \frac{C_i}{D_i} + \frac{C_s}{T_s} \leq (n + 1)(2^{1/(n+1)} - 1)$$

(note that this condition holds under the assumption that the deadlines of the periodic tasks equal their periods.)

Therefore, it must be:

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

Aperiodic guarantee of firm aperiodic activity $J_a$ (with known execution time $C_a = 1$) holds whenever the sufficient schedulability test is satisfied:

$$(1 + \left\lceil \frac{C_a}{C_s} \right\rceil)T_s \leq D_a$$

Assuming the integer valued parameters for the Polling Server, the smallest possible deadline $D_s = 42$, where the server’s parameters are determined by $C_s = 1$, $T_s = 21$ (since $1/21 < 0.048$, and a note that $[1/C_s] = 1$ for $C_s \geq 1$).

The next (higher) deadline than that determined just above is $D_a = 44$ (it corresponds to $T_s = 22$).