Inovace tohoto kurzu byla v roce 2011/12 podpořena projektem CZ.2.17/3.1.00/33274 financovaným Evropským sociálním fondem a Magistrátem hl. m. Prahy.

Evropský sociální fond
Praha & EU: Investujeme do vaší budoucnosti
Embedded and Real-time Systems
Response Time Analysis – Additions

http://d3s.mff.cuni.cz

Tomáš Bureš
<bures@d3s.mff.cuni.cz>
Response Time Analysis with Jitter

- So far we assumed that tasks are released at the exactly at beginning of a period
- What about if it is not so
  - For example due to inaccurate timer

- Release Jitter
  - the difference between the earliest and the latest time a task could be released relative to the start of the period

\[ J_i = J_i^{\text{max}} - J_i^{\text{min}} \]
May cause missed deadlines

<table>
<thead>
<tr>
<th>Task</th>
<th>T</th>
<th>D</th>
<th>C</th>
<th>R</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>30</td>
<td>20</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>B</td>
<td>1000</td>
<td>25</td>
<td>15</td>
<td>25</td>
</tr>
</tbody>
</table>

A is supposed to get activated here but it gets jitter of 10. B gets ready, but it has to wait for A. B gets preempted, Next instance. B misses deadline.
Response Time Analysis with Jitter

- How many times task \( \tau_2 \) may be preempted by \( \tau_1 \)?
  - \( n \) times if \( R_2 > (n - 1)T_1 - J_1 \)
  - Thus we need the biggest \( n \) satisfying condition

\[
\frac{R_2 + J_1}{T_1} > n - 1
\]

- That is:

\[
n = \left\lceil \frac{R_2 + J_1}{T_1} \right\rceil
\]

- Resulting into:

\[
R_i^{n+1} = C_i + B_i + \sum_{j=1}^{i-1} \left\lceil \frac{R_i^n + J_j}{T_j} \right\rceil C_j
\]
Now, we have to account for jitter of the actual task $\tau_i$

\[
 w_i^{n+1} = C_i + B_i + \sum_{j=1}^{i-1} \left( \frac{w_i^n + J_j}{T_j} \right) C_j
\]

\[
 R_i = J_i^{max} + w_i
\]

$w_i$ is the time taken for the task to complete once it has been released (i.e. a preemption window)
Response Time Analysis with System Overhead

- So far we assumed zero scheduling costs
- What brings the overhead?
  - Every preemption can result in two context switches ($C_{sw}$)
  - Timer clock tick ($C_{clk}$) – can be modeled as high-priority task
- Time to move a task from waiting to ready queue ($C_q$)

\[
 w_{i}^{n+1} = C_i + 2C_{sw} + B_i + \sum_{j=1}^{i-1} \left\lceil \frac{w_i^n + J_j}{T_j} \right\rceil (C_j + 2C_{sw}) + \sum_{\forall k \in all\_tasks} \left\lceil \frac{w_i^n}{T_k} \right\rceil C_q + \left\lceil \frac{w_i^n}{T_{clk}} \right\rceil C_{clk}
\]