

Aperiodic Scheduling

Earliest Deadline Due, Latest Deadline First, Earliest Deadline First

Exercise class 5

Presenter:
Jürgen Mattheis

In cooperation with:
Pascal Walter

Based on the lecture of:
Marco Zimmerling

December 6, 2022

University of Freiburg, Chair for Embedded Systems



Gliederung

Organisation

Overview Aperiodic Task Scheduling

Task 1

Task 2

Task 3

Literature

Organisation

Overview Aperiodic Task Scheduling



Overview Aperiodic Task Scheduling

	Deadline equals period	Deadline smaller than period
static priority	RM (rate-monotonic)	DM (deadline-monotonic)
dynamic priority	EDF	EDF*

Overview Aperiodic Task Scheduling

Schedulability test

	Deadline equals period ($D_i = T_i$)	Deadline smaller than period ($D_i \leq T_i$)
static priority	$\sum_{i=1}^n \frac{C_i}{T_i} \leq n \left(2^{1/n} - 1 \right)$ <p>(sufficient but not necessary)</p>	<p>(1) $\sum_{i=1}^n \frac{C_i}{D_i} \leq n \left(2^{1/n} - 1 \right)$ (sufficient but not necessary)</p> <p>(2) smallest R_i that satisfies</p> $R_i = C_i + \sum_{j=1}^{i-1} \left\lceil \frac{R_j}{T_j} \right\rceil C_j \text{ for all tasks } i$ <p>and $R_i \leq D_i$ (necessary and sufficient)</p>
dynamic priority	$\sum_{i=1}^n \frac{C_i}{T_i} = U \leq 1$ <p>(necessary and sufficient)</p>	<p>→ Buttazzo, <i>Hard real-time computing systems: predictable scheduling algorithms and applications</i></p>

Mixed Task Sets

- ▶ So far: we differentiated between **periodic** and **aperiodic** tasks.
- ▶ Now: Consider a **mixed** task set!
- ▶ We want to be able to find a schedule when there's both **periodic** and **aperiodic** tasks.

Schedulability tests

Sufficient? Necessary?

- ▶ We're interested in whether a given problem can be scheduled by algorithms.
- ▶ Depending on the algorithm we can derive sufficient and necessary conditions.

Sufficient: If $A \implies B$ then A is a sufficient condition for B.

Necessary: If $B \implies A$ then A is a necessary condition for B.

- ▶ A necessary and sufficient condition means, both statements are logically equivalent.

Schedulability tests

Utilization

Different kind of utilizations also play a big role in our analysis. We introduced the **processor utilization factor** $U = \sum_{i=1}^n \frac{C_i}{T_i}$ and later on U_s as the server utilization.

(More about servers later)

RM - Rate Monotonic Scheduling

Schedulability

- ▶ RM is optimal among all fixed-priority assignments in the sense that no other fixed-priority algorithm can schedule a task set that cannot be scheduled by RM.
- ▶ As in the lecture, we have $\sum_{i=1}^n \frac{C_i}{T_i} \leq n(2^{1/n} - 1)$ as a **sufficient** but not **necessary** condition.

RM(PS) - Rate Monotonic Polling Server

- ▶ One way to handle both periodic and aperiodic tasks is to use a so called server.
- ▶ This PS (Polling Server) acts as a periodic task (meaning it is instantiated at regular intervals T_s) whose job it is to, once it has the highest priority, serve any pending aperiodic requests within the limits of a server capacity C_s .
- ▶ Since we introduce yet another periodic task, the schedulability analysis simply is the same as normal *RM* with one additional task. Again, we have the **sufficient** but not **necessary** condition:
$$\frac{C_s}{T_s} + \sum_{i=1}^n \frac{C_i}{T_i} \leq (n+1)(2^{1/(n+1)} - 1)$$

Task 1



Task 1 I

Earliest Deadline First (EDF) and Total Bandwidth Server (TBS)

Task 1.1:

- ▶ what can be the maximum value of U_s such that the whole set (i.e. periodic tasks and the TBS) is schedulable with EDF?

Task 1 II

Requirements 1.1:

Schedulability test:

Given a set of n periodic tasks with processor utilization U_p and a total bandwidth server with utilization U_s , the whole set is schedulable by EDF if and only if

$$U_p + U_s \leq 1$$

► *processor utilization factor:* $U = \sum_{i=1}^n \frac{C_i}{T_i}$

Solution 1.1:

► *Maximum utilization of the Total Bandwidth Server:*

$$U_{s,\max} = 1 - U_p = 1 - \left(\frac{1}{3} + \frac{1}{5} + \frac{2}{13}\right) = \frac{61}{195} \approx 0.3128$$

Solution 1.2:



- First, we need to order the tasks by increasing release time r_i : J_4, J_6, J_5 . Then, we calculate the deadlines with $d_i = \max(r_i, d_{k-1}) + \frac{C_k}{U_s}$, where d_{k-1} denotes the previously calculated deadline ($k-1$ means the predecessor in the ordering according to the release time):
- $d_4 = \max(r_4, d_0) + 2/0.25 = 0 + 8 = 8$
 - $d_6 = \max(r_6, d_4) + 1/0.25 = 10 + 4 = 14$
 - $d_5 = \max(r_5, d_6) + 1/0.25 = 15 + 4 = 19$

Solution 1.3:

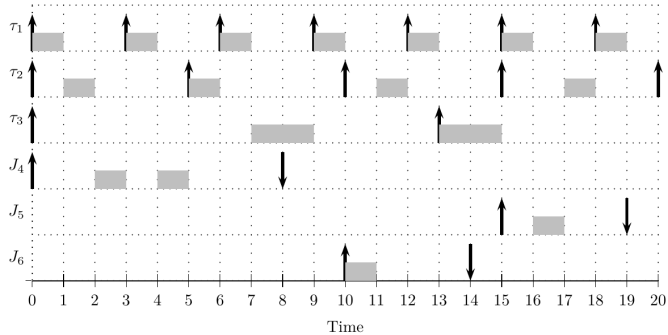


Figure 1: EDF schedule solution for Task 1

Task 2

Task 2 I

Schedulability Test for Fixed Priorities – Rate Monotonic (RM)

► asdff

Task 3

Task 3 I

Scheduling with Polling Server

► asdf

Literature

Bücher



Buttazzo, Giorgio C. *Hard real-time computing systems: predictable scheduling algorithms and applications*. Vol. 24. Springer Science & Business Media, 2011.