Networked Embedded Systems Lab
Prof. Marco Zimmerling

Introduction to Embedded Systems – WS 2022/23

Exercise 3: Aperiodic Scheduling

Task 1: Earliest Deadline Due

Check whether the Earliest Deadline Due (EDD) algorithm produces a feasible schedule for the following task set, given that all tasks are synchronous and arrive at time $t = 0$.

	J_1	J_2	J_3	J_4
C_i	3	6	2	4
D_i	8	15	3	11

Task 2: Latest Deadline First

Given the precedence graph in Figure 1 and the following table of task execution times and deadlines, determine a Latest Deadline First (LDF) schedule. Is this schedule feasible?

	J_1	J_2	J_3	J_4	J_5	J_6	J_7	J_8
C_i	3	4	2	3	3	2	2	1
D_i	5	8	11	15	12	18	19	20

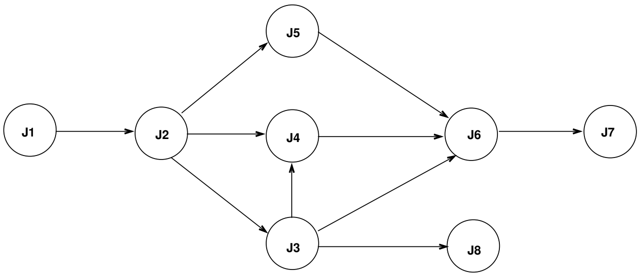

Figure 1: Precedence graph.

Task 3: Earliest Deadline First

In the following table, five tasks with arrival times, execution times and deadlines are given.

	J_1	J_2	J_3	J_4	J_5
a_i	0	2	0	8	13
C_i	3	1	6	2	3
d_i	16	7	8	11	18

(1) Determine a Earliest Deadline First (EDF) schedule. Is this schedule feasible?

(2) At time $t = 3$, a new task J_x arrives with execution time $C_x = 2$ and deadline $d_x = 10$. Can you guarantee the schedulability of the task set with this new task?

Task 4: Earliest Deadline First – Star

Given are seven tasks A, B, C, D, E, F, G with following precedence constraints:

$$A \longrightarrow C, \quad B \longrightarrow C, \quad C \longrightarrow E, \quad D \longrightarrow F, \quad B \longrightarrow D, \quad C \longrightarrow F, \quad D \longrightarrow G$$

All tasks arrive at time $t_0 = 0$, have a common deadline $d = 20$ and the following execution times:

	A	B	C	D	E	F	G
C_i	3	2	4	3	2	5	1

(1) Construct the precedence graph for this task set. Then, modify the release times and deadlines so that EDF* can be used for its scheduling.

(2) Determine a resulting EDF* schedule. For this schedule, compute the average of all response times of the tasks.

(3) Assume the additional precedence constraint $E \longrightarrow A$. Is there still a feasible schedule for the above task set? Justify your answer.

Task 5: Earliest Deadline First – Star

Given are eight aperiodic tasks, J_1 to J_8, with their arrival times, deadlines, and execution times as shown in the table below. Task precedence constraints are as follows:

$$J_1 \to J_2,\ J_2 \to J_3,\ J_3 \to J_4,\ J_5 \to J_6,\ J_6 \to J_7,\ J_6 \to J_8,\ J_2 \to J_7,\ J_7 \to J_4,\ J_8 \to J_7.$$

	J_1	J_2	J_3	J_4	J_5	J_6	J_7	J_8
r_i	0	3	4	0	0	2	0	2
d_i	3	8	15	15	10	10	10	11
C_i	1	3	3	3	1	1	2	1

(1) Construct the precedence graph.

(2) Using the EDF* algorithm, modify the arrival times and deadlines of the tasks in order to make the tasks schedulable under EDF. Enter the modified arrival times and deadlines in Table 1.

(3) Assume that the application is executed on a dual-core platform. At any time t, both cores execute the two ready tasks ($r_i^* \leq t$) with earliest deadlines (Note: A single task cannot be executed on two cores simultaneously). Using the arrival times and deadlines obtained in (2), construct an EDF schedule in Figure 2.

Table 1: Modified arrival times and deadlines

	J_1	J_2	J_3	J_4	J_5	J_6	J_7	J_8
r_i^*								
d_i^*								
C_i	1	3	3	3	1	1	2	1

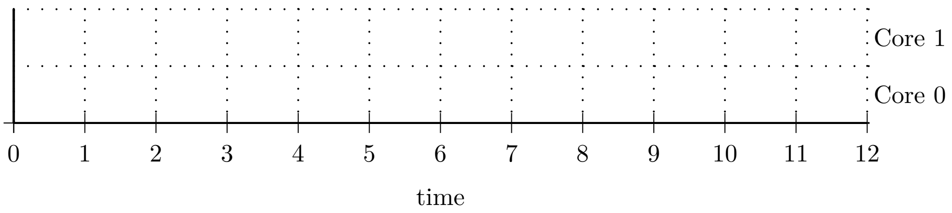

Figure 2: EDF schedule for part (3)

(4) Now assume that the application is executed on a quad-core platform with the same scheduling rule (4 cores execute the four ready tasks with earliest deadlines). Will executing on the quad-core platform reduce the completion time of the application? Justify your answer with an explanation.