https://folk.ntnu.no/michaeng/tdt4186_21/
michael.engel@ntnu.no

Theoretical exercises
Spring 2021

Theoretical Exercises 6

Real-time scheduling and OS

Please submit solutions on Blackboard by Thursday, 15.04.2021 12:00h

In addition to the lectures about embedded OSs, please also refer to the lecture on real-time scheduling!

6.1 EDF scheduling (3 points)

Suppose that we have a set of four jobs. Release times r_i, deadlines D_i, and execution times C_i are as follows:

- J_1: r_1=10, D_1=18, C_1=4
- J_2: r_2=0, D_2=28, C_2=12
- J_3: r_3=6, D_3=17, C_3=3
- J_4: r_4=3, D_4=13, C_4=6

Generate a graphical representation of schedules for this job set, using the earliest deadline first (EDF) scheduling algorithm.

6.2 Rate-monotonic scheduling (3 points)

Suppose that we have a system comprising two tasks. Task 1 has a period of 5 and an execution time of 2. The second task has a period of 7 and an execution time of 4. Let the deadlines be equal to the periods. Assume that we are using rate monotonic scheduling (RMS).

a. Could any of the two tasks miss its deadline, due to a too high processor utilization?

b. Compute this utilization, and compare it to a bound which would guarantee schedulability!

c. Generate a graphical representation of the resulting schedule! Suppose that tasks will always run to their completion, even if they missed their deadline.

6.3 Priority inversion (4 points)

Let A, B, and C be three tasks with priorities A=1 (highest), B=3, C=5 (lowest). Tasks A and C use a shared resource (e.g. shared memory) protected by a semaphore. The execution of the tasks is shown in fig. 1:

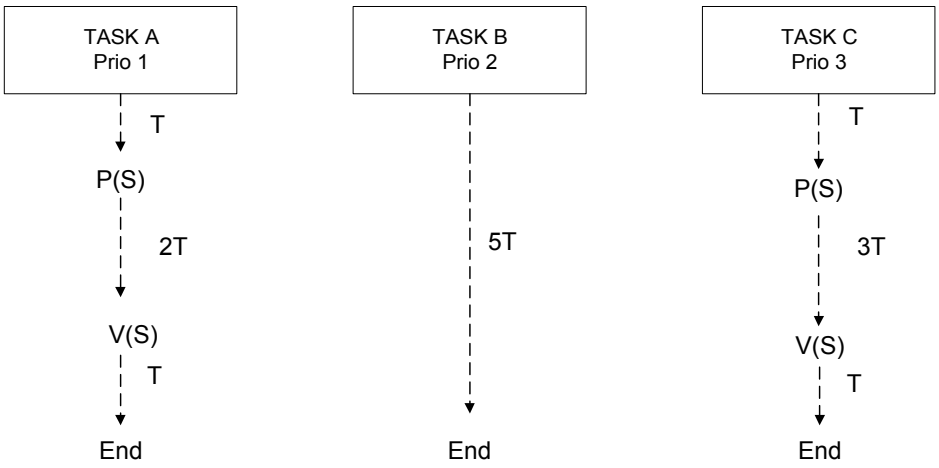

Figure 1: Execution of the tasks

The tasks are activated (once only) at the following time points:

- A: t = 2T
- B: t = 4T
- C: t = 0T

a. Use fig. 2 to visualize the *desired* execution sequence for the time interval $0 \leq t \leq 14T$.

b. Use fig. 3 to visualize the *actual* execution sequence with the according tasks states of the three tasks for the time interval $0 \leq t \leq 14T$.

c. Assume these are additional tasks with priorities between those of A and C which do not access the shared resource (no semaphores used). These tasks' individual priorities and execution times are not known. How long would the high priority tasks A be delayed in the worst case, then?

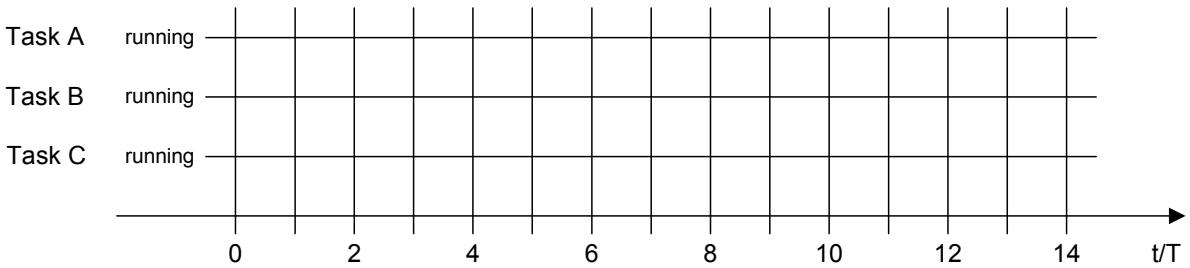

Figure 2: Desired execution sequence

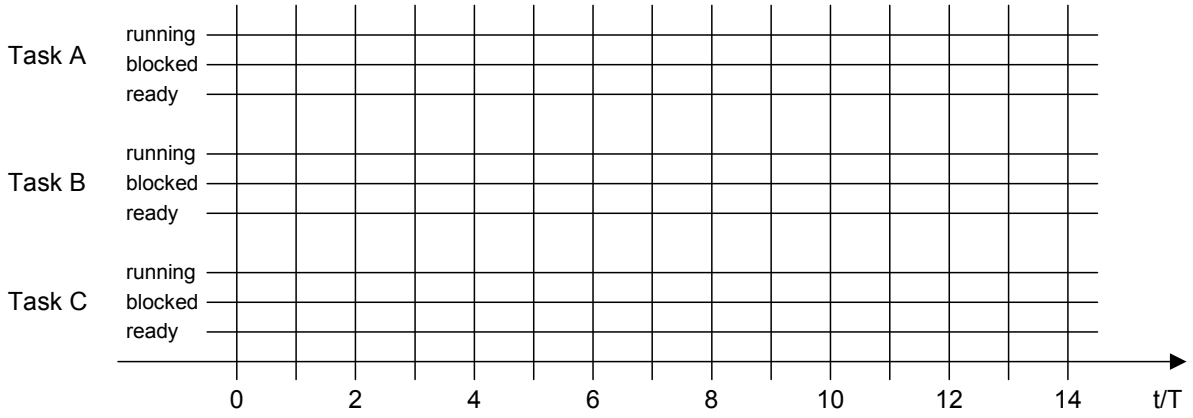

Figure 3: Actual execution sequence