RoboLab Real-Time Scheduling

Lab Report

# Preparatory Questions

## Question 1

Explain the mechanism behind the cyclic scheduling. What does cyclic schedule consist of?

The cyclic schedule consists of a major cycle which consists of several minor cycles. Processes are put into the minor cycles in the order they should be executed and the order of the minor cycles determine the overall order of the process executions. The schedule is created offline and uses static parameters.

What is the difference between a minor and a major cycle in a schedule?

The major cycle is repeated in infinity and consists of several minor cycles. By placing process executions in different minor cycles, you determine when a certain process should be run in the major cycle.

How are they determined?

The minor cycle is determined by the greatest common devisor between the processes periods. The major cycle is determined by the least common multiplier between the processes periods.

If the periods are not harmonic, then two different approaches can be taken. Either you push processes backwards in time when they clash or you run a process more often than it requires.

Outline the cyclic schedule for the RoboLab task set. What is the major and minor cycle?

The cyclic schedule we use in the lab is shown by Figure 1. The figure illustrates the major cycle with the first to fourth and the last minor cycle displayed. All minor cycles between the fourth and the last are equal. Each minor cycle is 50 ms and the major cycle is 1 s. This means that the major cycle consists of 20 minor cycles. The numbers on the planner process indicates the steps.

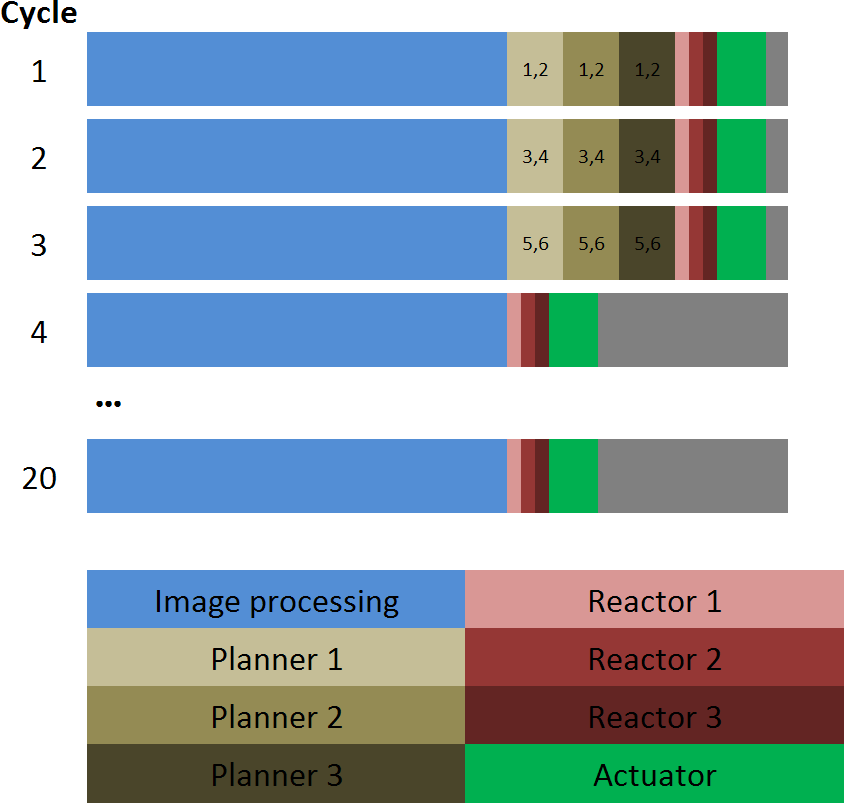


Figure 1: Cyclic schedule for the RoboLab task set.

When creating this schedule we have used 30 ms as WCET for the Image processing.

## Question 2

What is the difference between RMS and EDF scheduling with respect to implementation and performance?

RMS uses static priorities which are determined offline. The priorities are determined by the periods of the processes, longer period gives lower priority. EDF uses dynamic priorities and are therefore determined online. The priorities are determined by which process that has the closest relative deadline, closer deadline gives higher priority.

EDF requires more overhead at runtime and therefore RMS can be run faster. But EDF can manage to schedule task sets that RMS is not able to schedule because it gives a higher processor utilization.

## Question 3

Outline the steps in the RMS and EDF scheduling process, preferably with a sequence chart (or a pseudocode, or a flowchart).

The steps of the RMS scheduling process are shown by Figure 2 below. The boxes with dashed border represent interrupts.

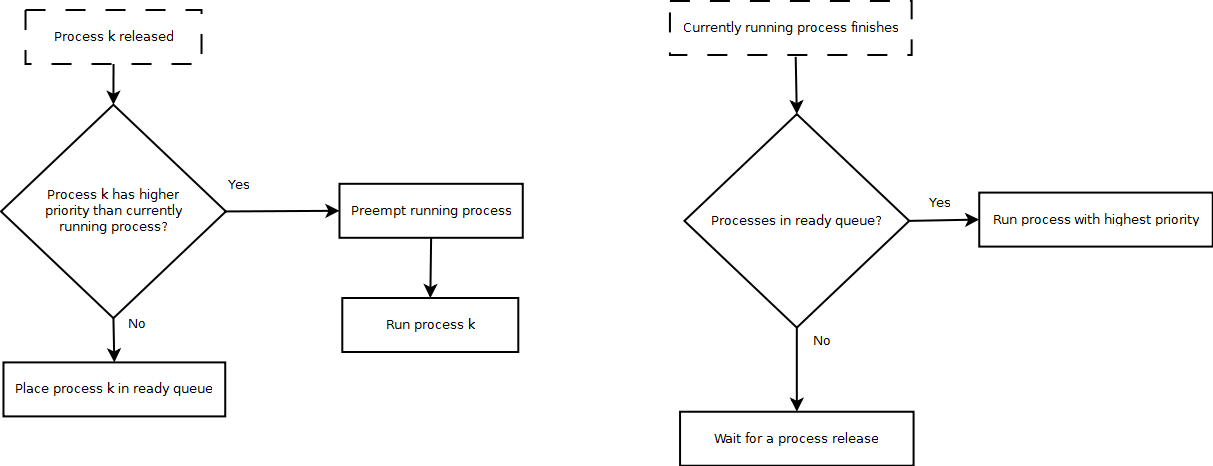


Figure 2: RMS flow chart.

The steps of the EDF scheduling process are shown by Figure 3 below. The boxes with dashed border represent interrupts.

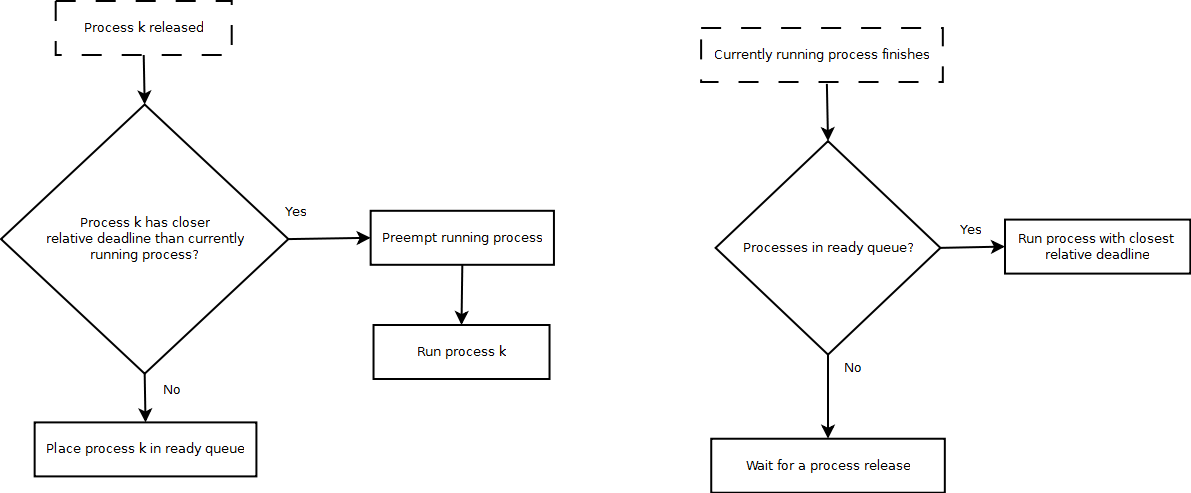


Figure 3: EDF flow chart.

What has to be done off-line (i.e. before you run the scheduler) and on-line (i.e. during run-time)?

When using RMS scheduling, you need to prioritize all processes/tasks offline. Online the scheduler only changes to the ready process with the highest priority as shown in Figure 2.

If EDF is used, nothing needs to be done offline. Online, the scheduler needs to prioritize the tasks in the ready queue. This is done by checking closest relative deadline. This is done when a task is released or has finished as shown in Figure 3. This is the only difference to a RMS scheduler.

## Question 4

Where and how do you look for deadline overruns in RMS and EDF?

The scheduler checks the deadline when the running process yields. It also checks the deadlines of the tasks in the ready queue when a new task should be started. In RMS, all tasks in the ready queue needs to be checked. In EDF, only the tasks up to a task which has not missed its deadline needs to be checked.

What has to be done when a deadline overrun has occurred?

The task needs to be stopped and rescheduled for its next deadline.

# Qualifying Questions

## Question 1

What can you conclude about the cyclic scheduling in dynamic environments?

Have you experienced any drawbacks or benefits of this type of scheduler in the RoboLab environment? Relate the theory on cyclic scheduling with your experience with the labs.

## Question 2

Compare the result of your simulations with the theoretical performance of the scheduling algorithms. Is the task set schedulable? How can the deadline miss ratio be explained? Where the results expected? Why or why not?

## Question 3

If you were the real-time system designer asked to implement a scheduler for an environment similar to RoboLab lab environment, what scheduling algorithm would you use? Motivate your answer based on your lab experiences.