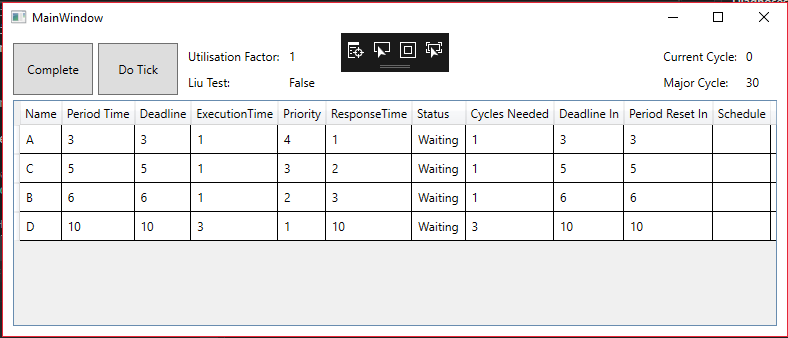
Ken Hasenbank & Artur Schmidt

# Lab 2

In this lab we should extend our implementation from the previous lab, so it is able to ‘calculate’ simplified and exact RTA. Another Task was to implement a timeline as a text string.

To run the Programm open the Projekt in Microsoft Visual Studio (2015 or higher). After running the program, there should appear the result in a window.



In the window you can see the Task, their period, procedure time and deadline. Based on this information and the used scheduling method the Priority will be assigned. A click on “Complete” will schedule the tasks, whereas a click on “Do Tick” will do it step by step.

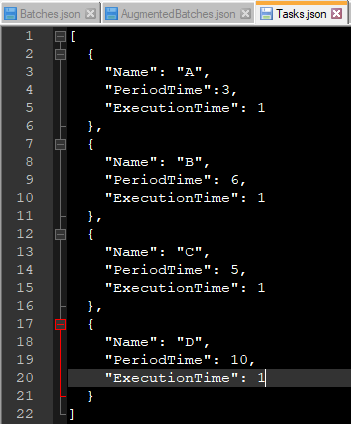
**“Cycles Needed”** displays how many cycles the Task needs to finish his tast. If the value reaches zero, the task is done and will wait until his period is over.

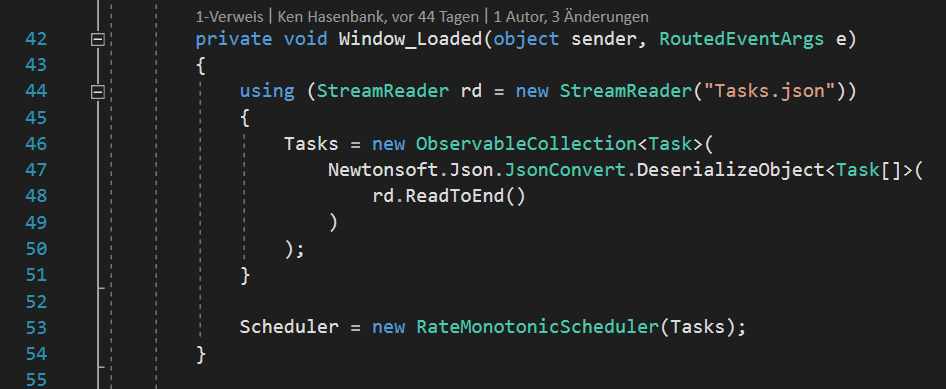
**“Deadline”** displays how many Steps are to go, until the task reaches the dead line. So the task should be completed within the remaining steps.

**“Period Reset in”** displays the time until a new period.

Change Values:

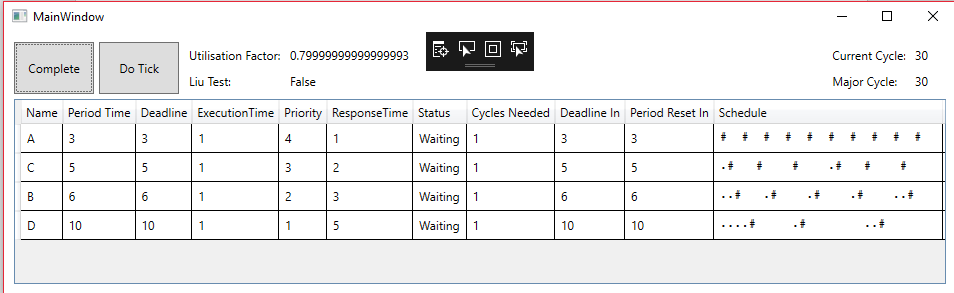
The User Interface is currently only to display the values and the schedule. It is not possible to change there any values. To changes the values (name, execution time, period, deadline) open *RTSScheduler\RTSScheduler\bin\Debug\Tasks.json.* The Program reads this JSON and uses the properties in it for the calculation.



To select the initial scheduling algorithm a short alternation to the code has to be done. In the MainWindow.xaml.cs in line 53 the desired Scheduler is spawned. 

Currently a DeadlineMonotonicScheduler and a RateMonotonicScheduler is implemented.

If the tasks are not schedulable, the algorithm for *optimal priority assignment* is used. Therefore, the priorities are swapped until a solution is found or not. If there is a possible solution, it will be immediately calculated, if not, a window appears and tells the user, that no solution exists.



When closing the Application a Output.txt will be generated which contains the calculated schedule with an additional Utilization line. The utilization shows which task is currently using the cpu.



