SAARLAND UNIVERSITY
COMPUTER SCIENCE

Prof. Dr. Reinhard Wilhelm
Dr.-Ing. Daniel Kästner
Dr.-Ing. Florian Martin
Sebastian Altmeyer, M.Sc.
Jörg Herter, M.Sc.

Development of Safety-Critical Embedded Systems
WS 2012/2013

Exercise Sheet 8

Please hand in the solutions to the theoretical exercises until the beginning of the next lecture, Fri. 2013-02-01, 10:00. Please write your name as well as the number of your tutorial group and/or the date/time slot on the first sheet of your solution.

Exercise 8.1: Deadline Monotonic Scheduling (Points: 6)

Consider the following properties of three independent tasks:

Task	Arrival Time	Execution Time	Deadline
T_1	0	8	24
T_2	0	12	18
T_3	0	4	12

Consider a preemptive real-time system consisting of one CPU.

1. Find a preemptive schedule, such that the tasks T_1, T_2, and T_3 finish within their deadlines. Draw the schedule and mark for each task, when this task owns the CPU.

2. Use **deadline monotonic** scheduling to schedule the three tasks. Draw the schedule and mark which (part of) task is using the CPU at each time point.

Exercise 8.2: Minimum Laxity First Scheduling (Points: 8)

Minimum laxity first scheduling activates at the current scheduling time t the task T_i with the smallest laxity. The laxity of the task T_i at the current time t is defined as

$$l(T_i, t) = d_i - t - r_i$$

where d_i is the deadline of the task T_i and r_i is the time remaining to complete the task T_i.

Consider the following properties of three independent tasks:

Task	Arrival Time	Execution Time	Deadline
T_1	8	12	36
T_2	10	4	30
T_3	16	10	28

Compute a preemptive schedule for one CPU for the given task set scheduled with minimum laxity first algorithm. Mark which (part of) task is using the CPU at each time point.