technische universität dortmund
fakultät für informatik
computer science 12

jian-jia.chen [☺] tu-dortmund.de
georg.von-der-brueggen [☺] tu-dortmund.de
wen-hung.huang [☺] tu-dortmund.de
jan.kleinsorge [☺] tu-dortmund.de

Exercises for Lecture
Real-Time Systems and Applications
Summer Semster 15

Exercise Sheet 5

(11 Punkte)

Exercise Due at Wednesday, June 10, 2015, 12:00 Uhr

Hinweise: Gruppenarbeit von bis zu drei Personen aus der gleichen Übungsgruppe ist möglich. Bitte vergessen Sie nicht Ihre Namen und Ihre Matrikelnummern auf die Lösung zu schreiben. **Die Abgaben können in den beschrifteten Briefkasten vor dem Sekretariat des LS12 (OH16/E22) eingeworfen oder per Mail (PDF Format) an georg.von-der-brueggen [☺] tu-dortmund.de abgegeben werden.**

Note: It is allowed to work in a group of up to three persons, if these persons are from the same practice group. Please do not forget to write your name and your Matrikelnummer on the solutions. **The solutions can either be placed in the mailbox in front of the secretary's office of LS 12 (OH/E22) or sent by mail (PDF format) to georg.von-der-brueggen [☺] tu-dortmund.de**

Exercise Sessions:

Do, 10:15 - 11:45	OH16/E18
Do, 14:15 - 15:45	OH16/E18

5.1 Resource Access Protocols (4 Punkte)

1. What is priority inversion? Is it possible to completely avoid priority inversion in fixed-priority scheduling? If yes, what is the drawback of such schemes? If no, explain your arguments.
2. Explain why Priority Ceiling Protocol (PCP) is deadlock free.
3. Mr. Smart wants to use PCP in his system which uses dynamic priority scheduling, i.e., EDF (earliest-deadline-first scheduling). Is it possible? What would be the problem(s) that he may face?
4. Explain how you will implement the PCP and Priority Inheritance Protocol (PIP) in a Real-Time Operating System.

5.2 PCP/PIP Schedulability Test (3 Punkte)

Consider the following case with four sporadic tasks and 3 semaphores, where $S_j(\tau_i)$ is the worst-case execution time of a critical section guarded by semaphore "S_j" in task τ_i and $S_j(\tau_i)$ is 0 when task τ_i does not need semaphore S_j.

	$S_1()$	$S_2()$	$S_3()$
τ_1	1	0	0
τ_2	0	0	9
τ_3	8	7	0
τ_4	6	5	4

	τ_1	τ_2	τ_3	τ_4
C_i	2	10	16	16
T_i	10	24	96	96
D_i	10	24	96	96

Suppose that the critical sections are not nested. Note that the worst-case execution time C_i of a task τ_i is derived by assuming that the critical sections are always granted without any blocking.

1. Can RM+PIP feasibly schedule the above task set?
2. Can RM+PCP feasibly schedule the above task set?

5.3 Non-Preemptive Scheduling (3 Punkte)

	τ_1	τ_2	τ_3	τ_4
C_i	2	3	2	3
T_i	9	12	18	24
D_i	9	12	18	24

1. Apply the sufficient utilization and blocking time based test by von der Brüggen, Chen, and Huang (slide 19 in 07-Non-Preemptive-Scheduling.pdf) to show that the task set is schedulable by non-preemptive rate monotonic scheduling.

2. Assume that due to a change in the implementation the WCET of τ_4 is increased to $C_4 = 4$. Is the task set still schedulable according to the test by von der Brüggen, Chen, and Huang?

3. Use the pessimistic schedulabulity test (slide 18 in 07-Non-Preemptive-Scheduling.pdf) that tests the schedulability for the preemptive and non-preemptive case in one equation to determine the schedulability of the new task set ($C_4 = 4$).

4. Why is it not always sufficient to only analyse the first release of each task if non-preemptive scheduling is used? Under what conditions is analyzing only the first release sufficient to determine schedulability? Explain why it was sufficient to only analyze the first release here.

5.4 Challenge on Optimal Priority Ordering (optional) (1 Punkt)

Suppose that the following schedulability test is an exact test for PCP: A system $\mathcal{T}$ of periodic, preemptable tasks with constrained dealines is schedulable on one processor by a fixed-priority scheduling algorithm if

$$\forall \tau_i \in \mathcal{T}\ \exists t \text{ with } 0 < t \leq D_i \ \ and \ \ W_i(t) \leq t$$

holds, where $W_i(t)$ of the task τ_i is defined as follows:

$$W_i(t) = B_i + C_i + \sum_{j=1}^{i-1} \left\lceil \frac{t}{T_j} \right\rceil C_j.$$

The worst-case blocking time B_i for task τ_i is at most

$$\max_{j>i,R} \{C_{j,R} | \Pi(R) \leq i\},$$

where $C_{j,R}$ is the worst-case (consecutively) execution time when resource R is required for executing a job of task τ_j. Please explain that rate-monotonic scheduling is an optimal fixed-priority scheduling policy under the above schedulability analysis.

General Hints: