technische universität dortmund — fakultät für informatik — cs 12 computer science 12

Prof. Jian-Jia Chen

Real-Time Systems (SS 2014)

Exercise 4: Resource Access Protocols and RT PL/OS

Discussion Date: 18, June, 2014

Exercise 4.1

(a) 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.

(b) Explain why Priority Ceiling Protocol (PCP) is deadlock free.

(c) 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?

(d) What are the advantages to use Real-Time Programming Languages and Real-Time Operating Systems to design real-time systems, respectively?

(e) Explain the concepts behind *Delay* and *Delay until* in Ada. Check the source code of FreeRTOS. What are the corresponding functions in FreeRTOS?

(f) Explain why most commercial Real-Time Operating Systems use only a fixed-length table to store the (pointers of) task control blocks (TCB). How does the scheduler in FreeRTOS work?

(g) Explain how you will implement the PCP and Priority Inheritance Protocol (PIP) in a Real-Time Operating System.

Exercise 4.2

Draw the current priority ceiling $\Pi'(t)$ of the system and the current priority of the jobs in the two examples of PCP (i.e., x axis with respect to time and y axis with respect to the priority levels) given in the lecture (i.e., Pages 25 and 26 in 06-Resource.pdf).

Exercise 4.3

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	8	6
T_i	10	24	48	50
D_i	10	24	48	50

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.

(a) Can RM+PIP feasibly schedule the above task set?

(b) Can RM+PCP feasibly schedule the above task set?

Exercise 4.4

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.