Errata Real-Time Architectures 2008/2009
Reinder J. Bril

Abstract
This document maintains a list of errors, typos, and other kind of mistakes that have been identified in the book, slides, or articles for the real-time architectures course (2IN25 and 2IN60) during the academic year 2008/2009. The errata that have been marked with a “*” gave rise to points to students.

1 Book

1.1 Section 4.3.1 “Optimality”
There is a typo in “Case 1” on page 80. Rather than “$C_1 < T_2 - FT_2$”, the relation should be “$C_1 \leq T_2 - FT_2$”. This is immediately clear from the top-line of the text on p. 81.

Note that
- the relation in “Case 1” on p. 82 is correct (and the text of both cases are identical, apart from the typo on p. 80).
- the relations in Fig. 4.5 (case (a)) and Fig. 4.6 reflect the actual schedules illustrated by the figures rather than the assumptions in the cases.

1.2 Section 4.4.2 “An Example”*
The reason for not being able to guarantee the schedulability of the task set under RMS in this example is wrong. A better, i.e. less pessimistic, premise would be $U > LL(2)$ and $HB(\tau_1, \tau_2) > 2$. Still, both tests are sufficient rather than exact tests. Using an exact test for RMS reveals that the task-set is indeed unschedulable under RMS.

Motivation: $U > \ln 2$ is the (lower) limit of the LL-bound as the number of tasks tends to infinity and should not be used to determine the schedulability of a task set consisting of a finite number of tasks.

Counter Example: Whereas the following task set with $\tau_1 (C_1=3, T_1=5)$ and $\tau_2 (C_2=1, T_2=7)$ is schedulable using RMS since $U (=0.74) < LL(2)$, it would appear not to be if the $U > \ln 2$ condition is used.

1.3 Section 5.6 “Sporadic Server”
The description of the SS algorithm is wrong; see also RTA.Exercises-5. A “proper” description of the SS algorithm can be found in Section 7.3 “Sporadic Servers” in [1].
2 Slides

2.1 RTA.B4-Policies-1 (SJF & LJF versus SJN & LJN)*

On slide 3, the terms Shortest Job First (SJF) & Longest Job First (LJF) are introduced. SJF has been developed primarily for batch-processing (see slide 4).

On slide 16, the terms Shortest Job Next (SJN) & Longest Job Next (LJN) are introduced as a priority assignment policy. Whether or not a scheduler using SJN and LJN pre-empts tasks is not mentioned on slide 16. For the exercise on slide 18, it was implicitly assumed that the tasks are scheduled pre-emptively.

Note that priority assignment and pre-emptive versus non-pre-emptive scheduling are two orthogonal dimensions of scheduling. For SJF and SJN, the same priority assignment is assumed. Similarly, the same priority assignment is assumed for LJF and LJN. The slides should be more explicit about the pre-emptive versus non-pre-emptive nature of the scheduler for SJF, SJN, LJF, and LJN.

2.2 RTA.Exercises-8+solutions

Slides 3 and 4 contain typos for HLP, PCP, and SRP:

- for HLP, task $\tau_2$ obviously raises its priority upon $L(r_2)$ rather than $L(r_1)$, and
- for PCP and SRP the system ceiling is raised when task $\tau_2$ executes $L(r_2)$, and task $\tau_1$ blocks on $L(r_1)$ rather than on $L(r_2)$.

3 Articles

References