Concurrent Programming: Principles and Practice

Errata Sheet for First Printing

page 20, line 3 of paragraph 3 — replace "welldefined" by "well defined"

page 28, 10th line from bottom — replace "resulting" by "result"

page 54, line 4 (exercise 1.23) — change G_1 to B_1 and G_n to B_n

page 74, process *Consumer* in Figure 2.3 — change loop guard to $c < n$ and change post-condition to $\{ IC \wedge c = n \}$.

page 84, equation (2.27) — change definition of Weak Fairness to:

A scheduling policy is weakly fair if (1) it is unconditionally fair, and (2) every conditional atomic action that is eligible is executed eventually, assuming that its guard becomes true and is not subsequently falsified, except possibly by the process executing the conditional atomic action.

page 84, equation (2.28) — change definition of Strong Fairness to:

A scheduling policy is strongly fair if (1) it is unconditionally fair, and (2) every conditional atomic action that is eligible is executed eventually, assuming that its guard is infinitely often true.

page 95, line 8 of paragraph 2 — insert "we" after "Since"

page 105, line 3 of equation (3.7) — insert **od** at the end of the **do** loop

page 127, Figure 3.17 — in Stage 3, the arrow from worker 6 should point to worker 4, not worker 1

page 162, exercise 3.9 — replace it by the one on the attached new page. I misunderstood the protocol in the cited paper, so asked the wrong questions.

page 186, line 7 — replace "say 0" by "say 1"

pages 187-189, Figures 4.10-4.12 — initialize *front* and *rear* to 1, not 0

page 228, second line above start of **Inference Rules** — replace "express" by "expressed"

page 229, second line above equation (5.1) — replace "The" by "This"

page 246, line 7 — replace "statement." by "statement,"

page 247, equation (5.9) — replace $d[i]$ by d_i and $b[i]$ by b_i

page 247, line 2 of second full paragraph — replace "showned" by "showed"

page 255, line 8 — replace [Dijkstra 1982] by [Dijkstra 1977]

page 272, line 4 — replace "Figure 6.2" by "Figure 6.3"

page 278, line 3 of equation (6.14) — replace "the the" by "the'

pages 282 and 283 — the first figure is 6.5 (a) and the second is 6.5 (b) page 321, second paragraph — change second sentence to "In particular, **empty** returns false if $nc > 0$, and it returns true if $nc = 0$."

page 328, line 2 of last full paragraph — replace "for specify permissible" by "for specifying permissible"

page 334, exercise 6.23 — replace the two occurrences of the phrase "only go around the tracks" by "go around the tracks only"

page 353, line 7 — in the assertion, replace *sent* [*ch*]1 by *sent* [*ch1*]

page 363, last line of *Client* in Figure 7.7 — change channel *release* to *request*

page 369, lines 4 and 5 — in the declarations of *open_reply* and *access_reply* , replace upper bound *n* by *m*

page 375, line 20 — delete **not** from the guard of the **if** statement

page 381, line 17 — replace *newto* by *newtop*

page 382, line 15 — replace *parent* by *first*

page 382, line 2 of *Initiator* process — replace *probe* by *probe_echo*

page 395, line 2 of paragraph 2 — change "some may" by "some nodes may". Technically, the term "cycle" should be replaced by "circuit" in this and the next three paragraphs.

page 409, line 8 of procedure *remote_send* — replace **dispatcher** by *dispatcher*

page 415, line 2 — replace "analyzes" by "analyze"

page 417, line 1 of exercise 7.13 — replace [Herman 89] by [Herman 1989]

page 444, both displays at the bottom of the page — replace **do** by **if** and **od** by **fi**

page 445, last line of first paragraph — replace "and and" by "and"

page 452, bottom of Figure 8.10 — replace $cij := aij * bij$ by $cij := cij + aij * bij$

page 455, lines 1 and 3 of *Client* process in Figure 8.12 — replace *server* by *serverid* in two places, and delete *serverid* from the parameter list of *open_reply*

pages 458 and 459 — replace the first three sentences of the paragraph that starts on the bottom of page 458 by:

The solution in Figure 8.14 is also fair, assuming that a philosopher does not eat forever and that a guard that succeeds is chosen eventually. In particular, if one waiter wants a fork that another holds, he will eventually get it. If the fork is dirty and in use, eventually the other philosopher will quit eating, and hence the other waiter will eventually execute the guarded communication statement that gives up the fork.

page 471 — replace it by the one attached. The original figure did not indicate what to do when no guard contains an output statement with a true boolean; it also did not reset variable *hc* in all cases.

page 485, first line of third display — insert *opname* after **proc**

page 510, last sentence of first paragraph — replace "service" by "serviced" page 528 — replace it by the one attached. The original *Worker* module did not enumerate all possible tours.

page 539, line 1 — delete " : **int** "

page 564, last line of second display — replace "*ch* ! *screen*" by "*screen* ! *ch*"

page 564, line 4 of second full paragraph — replace "process communicate" by "processes communicate"

page 579, lines 13 and 20 — replace "to" by "**to**" in the two for-all statements

page 580 — replace it by the one attached. The original *Worker* resource did not enumerate all possible tours.

page 587, line 11 of Figure 10.6 (b) — replace "< candidate" by "<= candidate" in the while loop guard

page 613, entry for Dijkstra, E.W. 1977 — change the title of the paper to "On two beautiful solutions designed by Martin Rem."

page 618 — add the following reference:

Kaashoek, M.F, Tanenbaum, A.S, Hummer, S.F., and Bal, H.E. 1989. An efficient reliable broadcast protocol. *Operating Systems Review 23*, 4 (October), 5-19.

page 621 — the page numbers for the first reference (McNamee and Olsson) are 357-387; the missing paper title in the second reference (Mellor-Crummey and Scott) is "Algorithms for scalable synchronization on shared-memory multiprocessors."