

Resit Concurrency & Multithreading

VU University Amsterdam, 15 January 2014, 18:30-21:15

(At this exam, you may use the textbook of Herlihy and Shavit. Answers can be given in English or Dutch. Use of the slides or a laptop is not allowed.)

(The exercises in this exam sum up to 90 points; each student gets 10 points bonus.)

1. Suppose a computer program contains a method that cannot be parallelized, and that accounts for 30% of the program's execution time. Use Amdahl's Law to determine the limit of the overall speedup that can be achieved by running the program on a multiprocessor machine. (8 pts)
2. Suppose that in the 2-thread consensus protocol using a FIFO queue (Figure 5.7 on page 107), a thread first dequeues an element from the queue, and then writes its value in its slot in the `proposed` array (i.e., lines 13 and 14 are swapped). Would the resulting protocol be wait-free? (Explain your answer.) (9 pts)
3. Consider the construction of a MRSW register from SRSW registers. Give (in detail) an execution in which one write by writer B and three reads by different readers A_i, A_j, A_k occur concurrently. Let the read by A_i return the old value, while the reads by A_j and A_k return the new value; moreover, let A_k read the old value on position `a.table[k][k]` of the array. Also discuss linearization points of your execution. (12 pts)
4. Give a scenario (including reads and writes to variables) of the CLH lock with timeouts in which initially the lock is free, four threads concurrently want to take the lock, two of the threads that are waiting for the lock concurrently leave the queue, the last thread in the queue is pointed forward twice, and ultimately this thread takes the lock. (10 pts)

5. Describe a lockless `contains()` method for the fine-grained linked-list implementation of sets. Show that your `contains()` method is linearizable (give linearization points, and consider the interplay with the `add()` and `remove()` methods). (12 pts)

6. Consider the work-stealing bounded queue. Give a scenario which shows that if the variable `bottom` were not declared volatile, then the algorithm would be incorrect. (12 pts)

7. Describe a variant of the termination detection barrier for the work-stealing bounded queue in which a thief communicates with the barrier not before it tries to steal a task, but after it has successfully stolen a task.
(Be careful that termination cannot be detected prematurely.) (12 pts)

8. Give a transactional specification, using atomic blocks, of the readers-writers lock. That is, when a writer is in its critical section, no concurrent readers and writers are in their critical section; but there can be concurrent readers in their critical section. Make sure that writers cannot be kept waiting indefinitely by a continuous stream of readers. (15 pts)