HW 3 GENERAL GRADING KEY & COMMON MISTAKES

Question 1

Section 1a:
A. “Mutual exclusion”, by violating the lock API by the user (e.g. unlocking not by the lock owner, or locking twice). Examples like this apply also on the valid implementations you have seen in class.
   -2 points
B. “Mutual exclusion”, one example for a reason given: “The comparison operation is not atomic” - it doesn’t have to be atomic at all and it doesn’t affect the atomicity of the lock. See the spinlock implementation from the lecture for example.
   -3 points
C. Missing “Progress” attribute
   -3 points

Section 1b:
A. Missing an answer to one of the questions (light/heavy critical section)
   -2 points
B. Missing ‘sleep’ as a solution to the busy-wait problem

Section 2:
A. Missing “Mutual Exclusion” attribute
   -4 points

Section 3:
A. “There is a race condition between the threads”
   -3 points
B. “There is only one processor, therefore the threads running sequentially”
   -4 points

Section 4:
A. Concurrent update of a value in the same register - can’t happen due to the context-switch mechanism, threads doesn’t share register values.
   -4 points
B. No possible-scenario example was given
   -3 points
C. Wrong minimal/maximal value
   -3 points
D. “The compiler translates this statement into 3 assembly instructions” - it doesn’t have to.
   This translation is one possible option out of many other op combinations and optimizations

Section 5:
A. “sum should be local variable” - the whole virtual address space, excluding memory mapped (using “mmap()”) objects, is duplicated exclusively for each process, not only the stack.
B. “The variable is shared, but…”
   -6 points
Question 2

Section a:
Mistake in the mutex_lock or mutex_unlock function that causes wrong behavior (-7).

Section b:
Because it is hard to thoroughly check this question, and since it’s a bonus question, students who attempted to implement this were given between 1 and 7 additional points, depending on how close to true, reliable and efficient their implementations were.

Section c:
Missing or wrong (-7)

Question 3

Section c:
Many students forgot to end the work of the consumers that are running endlessly in an infinite while loop. A possible solution for that would be a send a special “end” job, and if a consumer gets this special job, he will break from the loop (the special “end” job can be that we are simply sending NULL instead of a real job). (-2)

NOTE: This is a general correction key, the staff has the right to deduct points for your incorrectness in the solution as they fit.