

Assignment 9

Due: 1:25pm, Mon Dec 5th, 2022

Note: Make reasonable assumptions where necessary and clearly state them. Feel free to discuss problems with classmates, but the only written material that you may consult while writing your solutions are the textbook and lecture slides/videos. Solutions should be uploaded as a single pdf file on Canvas. **Show your solution steps** so you receive partial credit for incorrect answers and we know you have understood the material. Don't just show us the final answer.

This homework has an automatic penalty-free **1.5 day extension** to accommodate any covid/family-related disruptions. In other words, try to finish your homework by Monday 1:25pm to keep up with the lecture content, but if necessary, you may take until **Tuesday 11:59pm**.

1. Consistency Models (40 points)

For the multi-threaded program below, what are all the valid outputs (printed value of A) for a sequentially consistent execution of the program? Before these threads start executing, A and B are both initialized to zero. Assume that each line of code below corresponds to one assembly instruction.

Thread 1
B = 100;
A = 150;

Thread 2
if ((B+A) > 80)
then A = A - B;
print A;

What are possible valid outputs if the code in each thread is encapsulated within lock and unlock statements as shown below (again assuming a sequentially consistent processor)?

Thread 1
lock (L1);
B = 100;
A = 150;
unlock (L1);

Thread 2
lock (L1);
if ((B+A) > 80)
then A = A - B;
print A;
unlock (L1);

2. Consistency Models (20 points)

We discussed in class that high-performance techniques like out-of-order scheduling can yield unintuitive/unexpected outputs in multi-threaded programs written with a sequentially consistent model. We ultimately had a solution (a relaxed consistency model) that offered a relatively simple programming model and relatively high performance. To achieve this solution, briefly summarize what the programmer has to do, and what the hardware designer has to do.

3. Synchronization (20 points)

Briefly describe how LL-SC works by answering the following: (i) What happens when the LL instruction is executed? (ii) What happens when the SC instruction is executed? (iii) What is the benefit of using LL-SC instead of test-and-test-and-set?

4. Networking (10 points)

Why is the West-First routing algorithm better than Dimension-Order routing and Fully Adaptive routing?

5. Networking (10 points)

I'm trying to design a network topology for a system with 64 nodes. If my choices are a torus topology or a hypercube topology, what are the trade-offs that should guide my decision?
