Principles of Computer Systems
Final Exam

20-Dec-2018

This exam has 5 questions, totaling 100 points. You have 105 minutes to answer them, which means you earn about 1 point per minute of work – please consider spending on each question no more minutes than the number of points attributed to it.

If you exit the room during the exam, you will have to turn in your exam, and you will not be permitted to return to the room until the end of the exam. Please plan accordingly.

You are allowed to have any amount of printed material you like (books, papers, notes) but no laptops, tablets, cellphones, etc. are permitted during the exam. You must take the seat assigned by the course staff and present your CAMIPRO card to the staff upon request.

Do not open the exam until instructed to do so.

Your name: _____ SCIPER: _____

Question 1 *(10 points)*

Say you are designing a link-layer technology for a new type of wireless link, and you need to pick the maximum transfer unit (MTU) size. Describe the steps you would go through to choose the MTU.

Question 2 *(15 points)*

The TCP congestion control algorithm uses packet loss to decide when there is network congestion. A friend of yours says this is a layering violation.

(a) Is your friend right or not? Explain.

(b) Describe a scenario where relying on packet loss to decide when there is network congestion does not work well and explain why. What would be a better approach to do congestion control in the scenario you described?

Question 3 *(25 points)*

In class we discussed the Meltdown and Foreshadow attacks. A friend of yours says that the Meltdown attack demonstrates that Intel processors violate the atomicity of read operations promised in the Intel specs, because a process can see (and abuse) the side-effects of a read even if the read fails.

(a) What is (all-or-nothing) atomicity?

(b) Is your friend right or wrong? Explain.

Question 4 *(35 points)*

Consider a new low-cost CPU that has multiple cores, each core with its own cache, but no coherence/consistency guarantees (i.e., you may write to one cache on one core, and that value may never propagate on its own to the caches on other cores). A cache can be flushed in its entirety by privileged-mode code (i.e., by the operating system kernel). Say you want to run on this CPU a multi-threaded application in which data is shared between threads.

(a) Is it possible for the application to run correctly without any assistance from the OS (i.e., the OS does nothing to help compensate for the lack of cache coherence)?

(b) Is it possible for the OS to transparently provide to the application the illusion of a sequentially consistent memory (i.e., the memory hierarchy appears to execute all reads and writes to a single memory location in a total order that respects the program order of each thread)? In other words, could an application written under the assumption of sequentially consistent memory run on this OS and CPU correctly without any modification?

(c) What is the right balance of functionality between the application and the OS kernel when it comes to operating efficiently on this CPU? Propose a concrete system call API offered by the OS.

(d) Using the API you proposed above, sketch an implementation of a doubly-linked list that supports concurrent search and insert/delete operations from multiple threads simultaneously. (Ideally express the implementation in pseudocode, but you may describe it just in English as well)

Question 5 *(15 points)*

Name two of Lampson's hints that are reflected in the design of the Internet (viewed as a system) and explain how they have contributed to its success. If you can think of many that apply, pick the most interesting ones.