Virtual Memory and Caches Workout

Discussion 08 - 2 December 2019
CS 250P - Computer Systems Architecture
TA - Aftab Hussain

Problem 1. Explain how memory virtualization gives a process an illusion of a larger memory.

Problem 2. Explain how process isolation is guaranteed by memory virtualization.

Problem 3. Say we have a 4GB virtual memory address space for a process, and pages of 4kB each. How big is the translation structure going to be? Assume each entry in the translation structure is 4 bytes.

Problem 4. What's the difference between shared-memory programming model and message passing programming model?

Problem 5. What is the value of X in Memory at the end of Cycle 2, given the following sequence of events? Assume we are using a write-back policy.
Cycle 1: Value of X in Cache A is 0, Value of X in Memory is 0.
Cycle 2: CPU updates value of X in Cache B to 1.

Problem 6. Let us take the following sequence of events:
Cycle 1: Value of X in Cache A is 0, Value of X in Cache B is 0. Value of X in Memory is 0.
Cycle 2: CPU-A updates the value of X in Cache A to 1.

a. In the end of cycle-2, what would be the values of X in Cache B, and the memory if we are using a write-update cache coherence policy?
b. In the end of cycle-2, what would be the values of X in Cache B, and the memory if we are using a write-invalidate cache coherence policy?

Problem 7. Consider a symmetric shared-memory multiprocessor (3 processors sharing a bus) implementing a snooping cache coherence protocol. For each of the events below, explain the coherence protocol steps (does the cache flag a hit/miss, what request is placed on the bus, who responds, is a writeback required, etc.) and mention the eventual state of the data block in the caches of each of the 3 processors. Assume that the caches are direct-mapped and that each cache line only stores one word and that words X and Y map to the same cache line in each cache (in other words, X and Y cannot both be in the cache at the same time). At the start, X and Y are not in any of the three caches.

P1: Read X

P2: Write X

P2: Read X

P3: Write X

P1: Write X

P1: Read Y

P2: Read X

Reference: Anton's (UCI) and Rajeev's (Uni. of Utah) lectures on Computer Architecture