Homework #3 – 24 pts
Submit your homework to TEACH by Fri. 10/26/2018, at 11:59pm

Your submission should be comprised of one item: a .pdf file containing answers to the questions. Legible, handwritten solutions are acceptable for undergraduate students (472). The handwritten solutions must be electronically scanned and submitted as a PDF file. Graduate students (572) must compose their solutions in MS Word, LaTeX, or some other word processor and submit the results as a PDF file.

Clarification (10/24/18): As discussed in class on Tuesday, when I refer to 4KB or 16KB pages, I'm actually referring to 4KiB (4,096 bytes) and 16KiB (16,384 bytes). This will make the math easier.

1. (2 pts) For a 4K~~B~~ page, and a 32 bit address space, calculate the amount of memory needed to store a process's page tables. Assume each entry in the page table requires 12 bytes. Show all calculations.

2. (2 pts) For a 4K~~B~~ page, and a 64 bit address space, calculate the amount of memory needed to store a process's page tables. Assume each entry in the page table requires 12 bytes. Show all calculations.

3. (2 pts) For a 16K~~B~~ page, and a 32 bit address space, calculate the amount of memory needed to store a process's page tables. Assume each entry in the page table requires 12 bytes. Show all calculations.

4. (2 pts) For a 16K~~B~~ page, and a 64 bit address space, calculate the amount of memory needed to store a process's page tables. Assume each entry in the page table requires 12 bytes. Show all calculations.

Note: This next question is intended to give students a chance to demonstrate their understanding of caches. If you worked on homework 2 and understood the material, this should be a very quick question to complete. You can reuse a lot of your existing work.

5. (6 pts) Suppose the existence of an **associative cache design** with **N = 2.** Assume that the main memory uses a 32-bit address (and that each address maps to a single byte). If the cache scheme uses the following breakdown of the address bits:

Tag	Index	Offset
31-6	5-3	2-0

   a. What is the cache block size (in words)?
   b. How many block indices does the cache have?
   c. Including space for the valid bits, tags, and actual block data, how many bits would be required to actually implement this hypothetical cache?

6. (10 pts) Using the cache design from the previous problem, assume the following byte-addressed cache references are recorded. Assume that the cache is initially empty. Also assume that accesses occurred from left to right (e.g. address 0 was requested, then address 4, followed by address 16, etc).

Byte Address											
0	4	16	132	232	160	1024	30	140	3100	180	2180

   a. What was the number of cache hits?
   b. Create a table to show the final state of the cache including the value of each index, the valid bits, and the tags.