Assignment 4
Due date: Oct. 16th, 23:59

Each question is worth 2 points.

1. When using physical addresses directly, there is no virtual to physical translation overhead. Assume it takes 200 nanoseconds to make a memory reference. If we used physical addresses directly, then all memory references would take 200 nanoseconds each.

   a. If we use virtual addresses with page tables to do the translation, then without a TLB we must first access the page table to get the appropriate page table entry (PTE) for translating an address, do the translation, and then make a memory reference. Assume it also takes 200 nanoseconds to access the page table and do the translation. In this scheme, what is the effective memory reference time (time to access the page table + time to make the memory reference)?

   b. If we use a TLB, PTEs will be cached so that translation can happen as part of referencing memory. But, TLBs are very limited in size and cannot hold all PTEs, so not all memory references will hit in the TLB. Assume translation using the TLB adds no extra time and the TLB hit rate is 75%. What is the effective average memory reference time with this TLB?

   c. If we use a TLB that has a 99.5% hit rate, what is the effective average memory reference time now? (This hit rate is close to what TLBs typically achieve in practice.)

2. Consider a 32-bit system with 1K pages and simple single-level paging. With 1K pages, the offset is 10 bits.

   a. How many bits are in the virtual page number (VPN)?

   b. For a virtual address of 0xFFFF, what is the virtual page number?

   c. For a virtual address of 0xFFFF, what is the value of the offset?

   d. What is the physical address of the base of physical page number 0x4?

   e. If the virtual page for 0xFFFF is mapped to physical page number 0x4, what is the physical address corresponding to the virtual address 0xFFFF?

3. Suppose we have a computer system with a 44-bit virtual address, page size of 64K, and 4 bytes per page table entry.

   a. How many pages are in the virtual address space?

   b. Suppose we use two-level paging. How will the bits of the address be divided up? (each table at each level fits into a page frame)

   c. Suppose we have a 4 GB program such that the entire program and all necessary page tables (using two-level pages from above) are in memory. (Note: It will be a lot of memory.) How much memory, in page frames, is used by the program, including its page tables?

4. Consider the following page reference string:

   8, 7, 2, 9, 3, 1, 2, 5, 8, 3, 4, 6, 7, 7, 1, 0, 5, 7, 4, 6, 2, 3, 0, 1, 9

Assuming demand paging with four frames, how many page faults would occur for the following replacement algorithms: LRU replacement, FIFO replacement, Optimal replacement?

5. Assume that a program has just referenced an address in virtual memory. Describe a scenario in which each of the following can occur. (If no such scenario can occur, explain why.)

a. TLB miss with no page fault
b. TLB miss with page fault
c. TLB hit with no page fault
d. TLB hit with page fault (d is a bonus question -- 1 point)