CS 318/418/618, Fall 2022

Homework #4

Ryan Huang huang@cs.jhu.edu

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 will 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 TBL 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.

   a. With 1K pages, the offset is 10 bits. 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 a machine has the following hardware characteristics and current state

   (a) It takes 0.1 nanoseconds to access a cache line from its physical memory

   (b) It takes 1millisecond to access a page of 4Kbytes from the disk

   (c) It takes 0 second to access TLB, L1 & L2 processor caches (for simplicity)

   (d) All page tables can fit into the physical memory

   Suppose that we need to access the first word in page P, please list ALL possible latencies for this access and explain the reason for each case.

4. [Silberschatz] Consider a demand-paging system with the following time-measured utilizations:

   CPU utilization: 20%
   Paging disk: 97.7% (demand, not storage)
   Other I/O devices: 5%

   For each of the following, say whether it will (or is likely to) improve CPU utilization. Briefly explain your answers.

   a. Install a faster CPU

   b. Install a bigger paging disk

   c. Increase the degree of multiprogramming

   d. Decrease the degree of multiprogramming

   e. Install more main memory

   f. Install a faster hard disk, or multiple controllers with multiple hard disks

   g. Add prepaging to the page-fetch algorithms

   h. Increase the page size

5. If a large number of programs is kept in main memory, then there is almost always another ready program when a page fault occurs. Thus, CPU utilization is kept high. If, however, we allocate a large memory space to each of a few programs, then each program produces a smaller number of page faults. Thus, CPU utilization is kept high. Are these two arguments correct? Which policy, if either, should be preferred? Why?

6. [Crowley] 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 and arrange for all page tables to fit into a single page frame. How will the bits of the address be divided up?

   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?

7. [Crowley] Suppose we have an average of one page fault every 20,000,000 instructions, a normal instruction takes 2 nanoseconds, and a page fault causes the instruction to take an additional 10 milliseconds. What is the average instruction time, taking page faults into account? Redo the calculation assuming that a normal instruction takes 1 nanosecond instead of 2 nanoseconds.

8. If FIFO page replacement is used with a memory that only has four page frames, and there are only eight pages in the system (0-7), how many page faults will occur with the reference string 0172327103 if the four frames are initially empty? Now repeat this problem for LRU.

9. [Silberschatz] Consider the following page reference string: 7, 2, 3, 1, 2, 5, 3, 4, 6, 7, 7, 1, 0, 5, 4, 6, 2, 3, 0 , 1. Assuming demand paging with three frames, how many page faults would occur for the following replacement algorithms: LRU replacement, FIFO replacement, Optimalreplacement?