**Max Score = 15 points**

CS 250 2018 Spring Homework 09

This assignment is due at 11:59:00 pm Thursday, April 05, 2018.

Insert your typewritten answers into this file. You may include images of neatly hand drawn diagrams. To have this assignment graded, upload your file to Blackboard in either PDF or Word format. You may upload more than once to permit correction of errors. Late submissions will receive a score of zero (0).

You are responsible for ensuring that your upload (1) is to the location in Blackboard for this assignment, and (2) is the file that you intend to have graded for this assignment, and (3) is not marked “LATE” by Blackboard. You are encouraged to verify your upload was successful by downloading your file from Blackboard and examining that download.

Please be aware that Blackboard determines if an upload is late using an internal timer that measures time in units of seconds, but reports time to its users rounded down to the minute. This means that one second after Blackboard first shows time as 11:59pm any completing upload will be marked LATE because the internal program time is then after 11:59:00pm. Upload sufficiently before the stated deadline to avoid being late due to an upload that completes after 11:59:00pm.

1. A model for virtual memory performance is the average main memory access time, given by  Average\_main\_memory\_access\_time = Hit\_Ratio x Hit\_time + Miss\_ratio x Miss\_penalty, which is Equation 12.5 in the textbook.  Explain why replacing a given replacement algorithm with an algorithm that more closely approximates the optimal replacement algorithm will improve, have no effect, have an unknown effect, or will worsen each of the factors of the average main memory access time equation.

**There would be no effect on miss penalty because only a faster replacement algorithm would reduce the miss penalty, however in this case we have a optimal replacement which would create better quality replacements so there would be no effect. It would improve miss ratio because it would be a higher quality replacement algorithm and which would avoid future reloads of pages. It would improve hit time as well for the same reason as miss ratio.**

1. Assuming a page size of 16 Kbytes, what is the page number in hexadecimal and offset in hexadecimal for addresses 0x00000FFF, 0x00001FFF, 0x00002002, and 0xFFFFDFFD? Pad as needed with leading zeros to obtain hexadecimal representations for your answers.

|  |  |  |  |
| --- | --- | --- | --- |
| Given address | Binary address showing split | 0x Page number | Offset using 0x |
| 0x00000FFF | **000000000000000000 00111111111111** | **0x0000** | **0x0FFF** |
| 0x00001FFF | **000000000000000000 01111111111111** | **0x0000** | **0x1FFF** |
| 0x00002002 | **000000000000000000 10000000000010** | **0x0000** | **0x2002** |
| 0xFFFFDFFD | **111111111111111111 01111111111101** | **0x3FFFF** | **0x1FFD** |

1. A computer has 64-bit virtual addresses, byte-addressed memory, 1 Kbyte page size, 16 M page frames in main memory, and has 8 bits of metadata about each page.
   1. How many physical address bits are there in the address bus to main memory for this computer? **You would need 34 bits because with 16 M page frames with each page being 1 Kbyte it makes it need a total of 16 Gbytes of physical main memory which would need 34 bits to address.**
   2. A program for this computer uses a total of 8 consecutive pages of virtual memory to hold all of its instructions and data. How many bytes of main memory will be consumed for the page table of this program if the page table is single level? **There would be 2^56 bytes consumed because for 1Kbyte pages there is an offset size of 10 bits and with that 54 bits of virtual page number and 2^54 single page entries. Each entry holds 16 M page frames plus 8 bits metadata for a total of 32 bits. Putting all of it together the size would be 2^54 + 2^2 = 2^56 bytes.**
   3. Same question as part (b) but now the page table is a two-level design and the question is what is the lower bound for table space. The second level table holds the page metadata, not the first-level table. However, all page table entries are aligned in memory as 32-bit integers. **There would be 2^31 bytes.**
   4. Same question as part (b) but for a three-level table. **There would be 5 x 2^20 bytes**
   5. What are the ratios of sizes between all pairs of the three page table designs? Express your answers in the form x:1 with x rounded to the nearest integer.  
      **first:Second 2^56/2^31 33554432:1**

**First:Third 2^56/5x2^20 13743895347:1**

**Second:Third 2^31/5x2^20 409:1**