Assignment 8, Operating Systems

Jay R Bolton

December 15, 2011

Ch 8 problems: 8.1, 8.2, 8.5, 8.6, 8.11, 8.18

1. **8.1**
   (a) *a.* The CPU will add the root page table pointer and the virtual address' page table offset. The page table entry will be added with the virtual address' frame number offset to produce a physical address in main memory.
   (b) *b.*
      i. *(i)* Maps to VPN 1, which maps to PFN 7 with an offset of 28
      ii. *(ii)* Maps to VPN 2, which has no PFN, so page fault
      iii. *(iii)* Maps to VPN 5, which maps to PFN 0 with an offset of 379
2. **8.2** Did he make up the syntax with the semicolons inside the array brackets?
   (a) *a.* It's going to page fault every four iterations of the inner loop.
   (b) *b.* Swap j and i or swap the inner/outer loops.
   (c) *c.* This time it'll fault every four iterations of the outer loop – much much less often.
3. **8.5** Two page frames: [(A),(A,B),(C,B),(C,D),(A,D),(A,B),(E,B),(E,A),(B,A),(B,C),(D,C),(D,E)]
   Total transfers is the length of the list (12).
   Three page frames: [(A),(A,B),(A,B,C),(D,B,C),(D,A,C),(D,A,B),(E,A,B),(E,C,B),(E,C,D)]
   Total transfers is 9
   Four page frames: [[A],[A,B],[A,B,C],[A,B,C,D],[E,B,C,D],[E,A,C,D],[E,A,B,D],[E,A,B,C],[D,A,B,C],[
   Total transfers is 10
4. **8.6** Memory states with LRU replacement: [[1],[1,0],[1,0,2],[1,0,2,7],[1,6,2,7],[1,6,0,7],[1,2,0,7],[1,2,0,3],[
   [4,5,0,3],[4,5,0,1],[4,5,2,1],[4,5,2,6],[4,5,7,6],[2,5,7,6], [2,4,7,6],[2,4,7,3]]
   Total references is 33. Total transfers is 17. Memory hits are then 16 for a ratio of 16 to 33, which is about half.

Memory with FIFO replacement: 1,0,2,2,1,7,6,7,0,1,2,0,3,0,4,5,1,5,2,4,5,6,7,6,7,2,4,2,7,3,3,2,3
[[1],[1,0],[1,0,2],[1,0,2,7],[6,0,2,7],[6,1,2,7],[6,1,0,7],[6,1,0,2],[3,1,0,2], [3,4,0,2],[3,4,5,2],[3,4,5,1],[2,4,5,1],[
[3,2,7,4]]

Total references is 33. Total transfers is 18. Memory hits are then 15 for a hit ratio of 15 to 33, which is very slightly worse than LRU.

LRU was barely better in this case, so the extra overhead for implementing LRU over FIFO would probably not be worth it in this case if this access trace is indicative of most traces for this system.

5. **8.11** I assume probably double, so 400ns, but I couldn't find a specific reference to this data in the text.

   187ns. A miss will take a total of 420ns and a hit will take only 220ns.

   The TLB hit rate raises the EMAT quite a lot.

6. **8.18** We need to index the page and we need to have the offset. 5 bits can index 32 pages. An offset of 11 bits can index 2kb. This assumes that word length is 1 byte.

   The page table will need to index the data portion of memory. There can be 2^9 locations (2kb each) in memory, which can be indexed with 9 bits plus 1 for the "valid" bit, which makes for a total of 320 bits for the page table.

   Reducing memory by half will reduce the size of the page table.

   We'd have 2^8 locations instead of 2^9 so would have 9-bit page table addresses; the table's length would end up being 281 bits.