CS M151B Homework 8

Charles Zhang

March 1, 2022

Problem 5.16

As described in `Section 5.7`, virtual memory uses a page table to track the mapping of virtual addresses to physical addresses. This exercise shows how this table must be updated as addresses are accessed. The following data constitutes a stream of virtual addresses as seen on a system. Assume 4KiB pages, a 4-entry fully associative TLB, and true LRU replacement. If pages must be brought in from disk, increment the next largest page number.

Address:

Decimal	4669	2227	13916	34587	48870	12608	49225
Hex	`0x123d`	`0x08b3`	`0x365c`	`0x871b`	`0xbee6`	`0x3140`	`0xc049`

TLB:

Valid	**Tag**	**Physical Page Number**	**Time Since Last Access**
1	11	12	4
1	7	4	1
1	3	6	3
`0`	4	9	7

Page table:

Index	**Valid**	**Physical Page or in Disk**
`0`	1	5
1	`0`	Disk
2	`0`	Disk
3	1	6
4	1	9
5	1	11
6	`0`	Disk
7	1	4
8	`0`	Disk
9	`0`	Disk
a	1	3
b	1	12

a) For each access shown above, list

- **whether the access is a hit or miss in the TLB,**
- **whether the access is a hit or miss in the page table,**
- **whether the access is a page fault,**
- **the updated state of the TLB.**

b) Repeat `5.16.1`, but this time use 16KiB pages instead of 4KiB pages. What would be some of the advantages of having a larger page size? What are some of the disadvantages?

c) Repeat `Exercise 5.16.1`, but this time use 4KiB pages and a two-way set associative TLB.

d) Repeat `Exercise 5.16.1`, but this time use 4 KiB pages and a direct mapped TLB.

e) Discuss why a CPU must have a TLB for high performance. How would virtual memory accesses be handled if there were no TLB?

Problem 5.20

In this exercise, we will examine how replacement policies impact miss rate. Assume a 2-way set associative cache with 4 one word blocks. Consider the following word address sequence: 0, 1, 2, 3, 4, 2, 3, 4, 5, 6, 7, 0, 1, 2, 3, 4, 5, 6, 7, 0.

a) Assuming an LRU replacement policy, which accesses are hits?

b) Assuming an MRU (most recently used) replacement policy, which accesses are hits?

c) Simulate a random replacement policy by flipping a coin. For example, "heads" means to evict the first block in a set and "tails" means to evict the second block in a set. How many hits does this address sequence exhibit?

d) Describe an optimal replacement policy for this sequence. Which accesses are hits using this policy?

e) Describe why it is difficult to implement a cache replacement policy that is optimal for all address sequences.

f) Assume you could make a decision upon each memory reference whether or not you want the requested address to be cached. What impact could this have on miss rate?