ECE/CS 250 – Summer 2016 – Prof. Bletsch
Recitation #8
Caches and Virtual Memory

Objective: In this recitation, you will gain a greater understanding of caches, memory, virtual memory, and the intersection of caches and virtual memory.

Complete as much of this as you can during recitation. If you run out of time, please complete the rest at home.

1. Virtual Memory Examples

You have a 64-bit machine with 32KB pages. You have 1GB of physical memory.

(a) How many virtual pages does each process have?
(b) How many physical pages do you have?
(c) In a flat page table, how many PTEs do you have?
(d) If each PTE is 4B, how big would a flat page table be?
(e) Assume that the page replacement algorithm is LRU. How many distinct virtual pages would you have to see between accesses to a given page X for the second access to page X to be a miss?
(f) Sketch an 8-frame, 4-way set-associative TLB. How many bits is the "tag" (VPN)? How many bits is the "data" (PPN)? How big is the whole TLB?

2. Running out of Virtual Memory

On a 64-bit machine, it might appear you could never possibly run out of virtual memory. But what happens if you write a recursive program that (due to a bug) never reaches its base case? Try this. What happens?

3. Running out of Physical Memory

You can never actually run out of physical memory, but you can try to use more memory than you have, in which case the computer spends a lot of its time paging (servicing page faults). Open up the Task Manager to see how much memory you're currently using. Now start launching programs that use a lot of memory. What do you see in the Task Manager?

4. Caches and Memory – Putting it All Together

(a) Draw a complete memory system that includes the following: L1 I$ and L1 I-TLB, L1 D$ and L1 D-TLB, unified L2$ and L2 TLB, and main memory.

(b) Here are many of the possible outcomes for a given load. For each one, explain how it can occur and what happens in the given situation. Why is the shaded situation impossible?

situation	L1 D-TLB	L1 D$	L2 TLB	L2$	mem
1	hit	hit			
2	hit	miss		hit	
3	miss	hit			
4	miss	miss	hit	hit	
5	miss	miss	miss	hit	
6	miss	miss	miss	miss	hit
7	miss	miss	hit	miss	hit
8	miss	miss	hit	miss	miss
9	miss	miss	miss	miss	miss

(c) What are the trade-offs between handling a TLB miss in software versus handling a TLB miss with hardware? Consider issues like latency, hardware cost, backward compatibility, etc.

(d) Let's say you want to implement a virtual/physical cache (virtually indexed and physically tagged). Assume: 32-bit architecture, 16B cache blocks, 32KB pages. The cache is 128KB. How set-associative must the cache be to permit the use of virtual indexing with physical tagging?