

ECEn 528

Study Guide – Virtual memory and virtual machines

- Read Section B.4-B.5, 2.4 of H&P
 - Things to focus on
 - How virtual memory works
 - TLBs
 - Paged virtual memory (in section B.5)
 - Don't worry too much about the details of segmented virtual memory
 - Clarifications
 - None
 - Answer the following questions:
 1. On page B-12 (back in section B.1), the text indicates that the Opteron's memory hierarchy uses a 48-bit virtual address, even though the processor can generate 64-bit virtual addresses. What advantages and disadvantages result from this decision?

The smaller size simplifies virtual address mapping: a larger virtual address space implies a larger page table

A larger size would allow more memory to be addressed, but this isn't really needed (yet)

2. What disadvantages are there to avoiding address translation when accessing a cache?
How can you overcome these disadvantages?

Avoiding address translation is faster

3. How are virtual memory systems similar to caches? How are they different?

They both provide a mapping between resources, but caches are used to reduce access time to slow main memory/disk, whereas VM is used to allow multiple processes to run

4. Why are multi-level page tables used?

They reduce the overall size of the page tables, at the cost of longer translation times

5. What are the advantages and disadvantages of inverted page tables?

Inverted page tables allow the size of the table to be proportional to physical memory, not the virtual address space
An inverted table provides only one forward mapping, so it's hard to share memory among processes

6. Assume 7-bit virtual addresses, 12-bit physical addresses, and 16-byte pages. Given the following page table, what is the physical address corresponding to virtual address 0x65?

	<i>PPN</i>
0	0x48
1	0x55
2	0x37
3	0xa5
4	0xc7
5	0xf3
6	0x01
7	0x34

VA = 0x65 = 110 0101

page number = 110 = 6 -> 0x01

page offset = 0101 = 5

physical address = 0x015

7. What instruction set features can make it difficult to write a VMM?

Instructions that read control registers in user mode and reveal the guest OS is running in a VM

Instructions that check protection but assume the OS is running at the highest privilege level