Paging and TLB (Ch 18 & 19)
Discussion Questions

- What is the problem with segmentation that paging attempts to solve?
  - external fragmentation
- What is the difference between a page and a page frame?
- What is the equivalent of base and bounds registers for paging?
  - page table
- How many of these page table exist?
- Is address translation easier or harder in paging than segmentation?
- What does a page table entry store?
  - VPN => PFN
  - valid bit
  - protection bits
- How does the valid bit help support sparse address spaces?
- Why do we need protection bits for PTE?
- How does the CPU find the page table for a given process?
- What are the two primary problems with paging as presented in this chapter?
  - slow: two physical accesses per virtual access
  - large amounts of phy. memory used to store page tables
- What is a cache?
  - what are they used for?
  - how does caching make address translation faster?
- Where is the TLB located?

- Who handles a TLB miss?
    - on x86 and ARM (hardware)
    - on Sun Sparc (OS)
- What if software managed TLB handling code is on a page that is not in the TLB?
    - how can the OS avoid this?
- In architectures without ASID, how does the MMU deal with multiple TLB entries with the same VPN?
    - what effect does this have on context switching?
- How do ASIDs solve this problem?