Summary

• We know how address translation works in the "vanilla" case (single-level page table, no fault, no TLB)
  – hardware splits the virtual address into the virtual page number and the offset; uses the VPN to index the page table; concatenates the offset to the page frame number (which is in the PTE) to obtain the physical address

• We know how the OS handles a page fault
  – find or create (through eviction) a page frame into which to load the needed page
  – find the needed page on disk and bring it into the page frame
  – fix up the page table entry
  – put the process on the ready queue

• We're aware of two "gotchas" that complicate things in practice
  – the memory reference overhead of address translation
    • the need to reference the page table doubles the memory traffic
    • solution: use a hardware cache (TLB = translation lookaside buffer) to absorb page table lookups
  – The memory required to hold page tables can be huge
    • solution: use multi-level page tables; can page the lower levels, or at least omit them if the address space is sparse
    • this makes the TLB even more important, because without it, a single user-level memory reference can cause two or three or four page table memory references... and we can't even afford one!

The TLB

• Implemented in hardware
  – fully associative cache (all entries searched in parallel)
  – cache tags are virtual page numbers
  – cache values are page table entries (page frame numbers)
  – with PTE + offset, MMU can directly calculate the physical address

• Can be small because of locality
  – 16-48 entries can yield a 99% hit ratio

• Searched before the hardware walks the page table(s)
  – hit: address translation does not require an extra memory reference (or two or three or four) – "free"
  – miss: the hardware walks the page table(s) to translate the address; this translation is put into the TLB, evicting some other translation; typically managed LRU by the hardware

• On context switch
  – TLB must be purged/flushed (using a special hardware instruction) unless entries are tagged with a process ID
  – otherwise, the new process will use the old process's TLB entries and reference its page frames!