1. Recap of example from last time
    a. 12-bit virtual addresses, page size of 256 bytes, physical memory size of 2 KB
    b. Layout of address below

Virtual Page Number (VPN)	**Offset into Virtual Page**	=	**Virtual Address Size**
4 bits	8 bits		12 bits

Physical Frame Number (PFN)	**Offset into Physical Frame**	=	**Physical Address Size**
3 bits	8 bits		11 bits

    c. Translated the virtual address 0x716
        i. Corresponding physical address was 0x216
    d. State of memory and page table after the previous translation

Memory

Physical Frame Number	**Size**	**Address Range**	**Status**
000	256 B	0x000 – 0x0FF	Busy
001	256 B	0x100 – 0x1FF	Busy
010	256 B	0x200 – 0x2FF	Process A
011	256 B	0x300 – 0x3FF	Busy
100	256 B	0x400 – 0x4FF	Empty
101	256 B	0x500 – 0x5FF	Process A
110	256 B	0x600 – 0x6FF	Busy
111	256 B	0x700 – 0x7FF	Empty

Page Table

Virtual Page Number	**Status**
0000	On Disk
0001	Frame 101
0010	On Disk
0011	On Disk
0100	On Disk
0101	On Disk
0110	On Disk
0111	Frame 010
1000	On Disk
1001	On Disk
1010	On Disk
1011	On Disk
1100	On Disk
1101	On Disk
1110	On Disk
1111	On Disk

2. Page table access example – hit
    a. Now assume we're given the next virtual address
        i. 0x15F = 0001 0101 1111
    b. Look in the page table for the VPN 0001
        i. Page table tells us the frame is in memory, and the PFN is 101
    c. Translate address using the PFN
        i. Virtual address 0x15F = 0001 0101 1111
            1. Replace first 4 bits with 3-bit sequence 101
            2. Offset stays the same
        ii. Thus, physical address = 0x55F = 101 0101 1111
    d. Now that we have the physical address, go out to cache/RAM to retrieve data at that location

3. Translation lookaside buffer (TLB)
    a. Need to make these translations extremely often
    b. Instead of constantly going to RAM for page tables, cache recent translations from VPN to PFN

   c. Cache needs to have extremely high hit rate, so use FA cache
      i. Various other techniques can increase the hit rate, too
   d. Hit in TLB? Great, immediately have translation ready
   e. Miss in TLB? Must go to page table to obtain translation (or find out there's a page fault)

4. Virtual memory summary
   a. Virtual memory is a mapping from the process' virtual space to the real/physical RAM space
   b. Need to translate virtual addresses from the currently running process to physical addresses in RAM
   c. Page table tells us exactly how to make that translation, and if page is currently in RAM or not
   d. Can have more memory for one process than the amount of RAM we have
   e. Currently unused frames are stored on disk, and brought out when needed
   f. TLB is used to cache recent virtual to physical translations