Lecture 30
July 21

Virtual Memory

- Virtual memory is a large amount of "memory"
    - Doesn't really exist
    - The programs see this large virtual memory because the MMU creates the illusion
- The contents of the VM are broken into pages.
    - Each page is (at any given time either:
        1. in physical memory
        2. on the hard disk (in swap space)
        3. unused
- The MMU makes this look like one large piece of RAM.
    - Must translate virtual → physical addresses (case 1)
    - Must generate page faults & have OS swap (case 2)

Page Tables

- A virtual page could be stored any where in physical RAM
- The MMU & OS must keep track of where pages are stored
    - If the pages are in memory, the MMU must know which physical page
    - A "page table" stores this
- The page table entry for each virtual page will contain (at least):
    - Physical page frame: which page in memory holds this virtual page? (if it's in memory)
    - Validity bit: is it in memory??
        - I.e. is the physical page frame valid??
    - Dirty bit: has the page changed since it was last on the disk?
        - If not, it doesn't have to be written out again.
    - Used bit: is this virtual page used?
- The page table will be stored in RAM
    - Every memory access a program makes will now require two: read page table, read data
    - Parts of the page table could be stored in memory cache
- If the processor is pipelined this could be less of a problem
    - The instruction fetch & operand fetch could be two stages each.
- Since the page table is accessed so often, it could be cached separately
    - The "translation lookside buffer" (TLB)
    - A cache of: virtual page number
    - Corresponding physical page
    - Valid & dirty bits

Input-Output (ch 11)

- Computers need to communicate with the peripherals.
    - Peripherals: input or output devices that is connected to the computer.
    - E.g. keyboard, display printer, hard drive

I/O interfaces

- I/O interface: a way to connect I/O peripherals to the CPU
    - Usually a bus interface
    - The I/O interface has to:
        - Convert signals to the right form (positive/negative logic, voltage, etc)
        - Synchronize input signals
        - Data conversion (code conversion)
        - Control of the peripherals