8/20 Day 1

Course Outline:

- Hardware — 2 weeks
- Process managment — first test
- Memory managment —
- Scheduling —
- Input/output managment —
- File systems — Out of time

Hardware

- OS job is to dictate hardware what to do.
- John Von Neuman
  - what a computer program should look like.
  - we needed 5 elements of modern computers
    - CPU
    - Input
    - output
    - Memory
    - Mass Storage

CPU - responsible for the execution of machine instructions

- must fetch instruction from memory.
  - execute it
- three parts to CPU
  - controll unit
    - normal instruction execution process
    - interrupts and traps
      - Printer out of paper
      - Traps are errors caused by software. $(\frac{x}{0})$
    - privedeged and unprivledged mode
    - interval timing
  - ALU - Arithmetic logic unit
  - Registers
    - local storage for CPU
    - size measured in bits
      - typically same size as instruction 32 bit 64 bit
      - data registers
      - Program counter
        - Address of next instruction.
      - Instruction register
        - Current instruction
      - Stack pointers
        - used with subroutines,
      - Relocation register

- calculating memory addresses
- from virtual to physical.

- Program Status word
  - contains status information about instruction
  - like the sign bit (+, -)
  - 0 set when last op = 0
  - carry - indicates a carry from last process
  - equal
  - overflow
  - interrupt enable / disable
  - supervisor

Continueing Hardware

ALU - Arithmetic logic unit

cpu's job is to execute instruction cycle

- cpu
  - Instruction cycle
    - uninteruptable

Steps

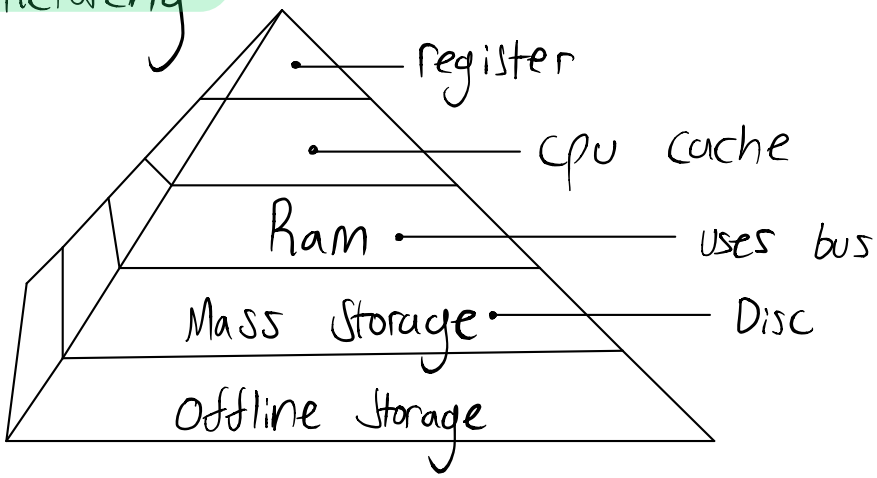
1. Check for interrupts
   - if thoe are, handle them
2. fetch next instruction from memory into instuction register
3. increment the program counter
4. determine type of instruction just fetched
5. if instrustiction requires data find it.
6. fetch found data into data register
7. execute the instruction
8. Store results in proper location
9. return to step 1.

Note: Windows system idle process does nothing but allow the system to go into this process

- Types of machine instructions
  1. process ⟺ main memory - move data
  2. processor ⟺ I/O device
  3. data processing
  4. control - may specify that the sequence of the instructions be altered

- Memory
  - bit - single binary option
  - byte - 8 bits, size of cpu register way back when
  - nybble - half a byte, 4 bits
  - word - native register size
  - double word - twice size of word
  - quadword - twice the size of double word

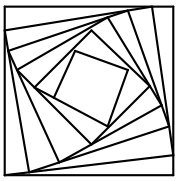
Memory Hierarchy



- if you have a 64 bit os you need a 64 bit bus.
- A Bus is a topology type - A half duplex system

- Seek time - finding track where information is
- disk latency - rotational delay to read information

- Input / output processing
  - four techniques in use
    1. Programmed I/O
       - Uses single instruction for input and output, for each "word" of data transfered.
       - Really inefficient - waiting for device.
    2. Interrupt driven I/O
       - Start I/O instruct.
       - tell peripheral to generate an interrupt when finished.
       - Cons:
         - Still one instruction / word of data moved
         - one interrupt / word of data moved
    3. Direct memory access - DMA : used on PCs
       - Programmed I/O performed by device controller
       - Cpu tells controller to move entire block of data
       - DMA does programmed I/O to move block of data
       - Issues an interrupt when finished
       - Con: DMA and CPU have to share Bus cycles
    4. Data channels - Main frame computer
       - dedicated processor for I/O with its own memory bus



- Polling
  - where a query is sent to the device to determine the device status.
- Interrupts — modern machines are interrupt driven
  - A signal from a device that asks another device or the operating system to a change in status

- Interrupt processing — Look up
  - process begins in hardware
    1. interrupting device controller
       - interrupt on interrupt line
    2. When cpu is ready the handle interrupt
       - it sends an acknowledgment back to device
    3. when device controller gets it sends back an int to identify the interrupt called.
    4. the cpu removes interrupt vector function
       - Saves it some where temporarily
    5. Cpu pushes program counter and program word on stack
    6. Cpu obtains new program counter
       - by using interrupts vector as an index to pc value

- Then goes to ~~hardware~~ Software

(7) interrupt service routine save all CPU registers in the register save area.

(8) determin which device issued interrupt

(9) Service deals with cause and interrupt

(10) if needed alert device that interrupt was serviced

(11) CPU registers restored

(12) Program count & PSW are restored

(13) CPU continues as if nothing happened.

Note: the interrupt handler cannot be interrupted

- Ctrl + alt + delete - is a user generated interrupt.

Types of interrupts

(1) program interrupts
- result of instruction execution
- error messages
- Traps
  - Similar to interrupt, but produced by software

(2) Timer interrupts

(3) I/O interrupts
- caused by controllers

(4) Hardware failure
- Disk read error

NOTE: need to know system cycle + Interrupt processing

Hw 1. Chapter 1

5, 6, 7, 8, 9, 10, 12, 13

- typo on #9 $106 = 10^6$

Chapter 2

- Operating System
  - Set of programs implemented in software and firmware that are designed to make hardware useable
  - ▶ Reason for operating system
    1. Provide a more reliable and hospitable interface to the hardware
    2. Provide for a more efficient use of the hardware to increase productivity and reduce costs
    3. Provide for controlled sharing of resources
       - cpu cycles
       - memory
       - files
       - storage
       - printers

Performance Goals of OS

1. Provide high utilization of resources to keep costs down
2. Should provide high availability of resources
3. Promote user productivity
   - Make sure user works well
4. Reliable
5. Provide protection for important resources.
6. efficienty
7. predictable
8. easy to use
9. Provide system resources