**CPSC 457 – Lecture #2** May 11, 2016

**Today we are going to cover:**

* Operating system history
* Hardware Review
* Assignment #1

# Operating System History

**Tanenbaum’s Generations**

1. Vacuum Tubes (1945-1955)
   1. Nobody knew about computer
   2. Created in World War I
2. Transistors & Batch System (1955-1965)
   1. More general purpose of computing
   2. Size of computer was really big
3. Integrated Circuits & Multiprogramming (1965-1980)
   1. Start writing programming
4. Personal Computers (1980 – Present)
   1. More manageable size
   2. Multiple programs are running
   3. Example: Windows OS
5. Mobile Computers (1980 – Present)
   1. Our current mobile phones

Babbage

* Responsible of fixing tables
* In early 1900
* He developed mechanical computer, able to provide polynomial equation and give tables
* Good news: it kind of worked
* He also developed Analytical Engine – purpose of multi calculation
  + Youtube Video: “A demo of Charles Babbage’s Difference Engine”

1st Generation

* Harvard built IBM
* First computer built in England called EDSAC
  + Youtube Video: “Inside the Computer (EDSAC) – Computerphile”
* Very slow due to mercury
* Each machine is unique
* They blow up often

2nd Generation

* Possible to sell computer to customers
* Programmer: read and write in paper and generate punch cards
* Batch processing: it is seller able but expensive
* IBM 7094 – bad for calculation / good for I/O
* IBSYS: first OS

3rd Generation

* Able to taxes, item inventory and more
* Computer becomes smaller
* IBM 360 : flagship computer
* Multiprogramming: start use CPU (doing more than one thing at once)
  + Less waste time in CPU
  + Got better for programmers but still slow
* Timesharing: when is active, active people get time, non-active people don’t get time
* MULTICS: Multiplexed Information Computing Service
  + Becomes faster
  + PL1 – failed due to they did not build compiler
  + PDB
* Ken Thompson
  + He wrote Unix – 1969
  + Code was available for public
  + Microsoft – Xenix (own version of Unix)
  + Important of UNIX
    - First programming language called B
    - Better version is C
    - Common set of tools (Ex: ls)

4th Generation

* Size become something you can sit with desk
* First computer created from hobbyist community
* IBM PC came out – Monitor, keyboards etc...
  + IBM realized they need OS
  + CMP rejected selling OS
  + Contacted Bill Gates
  + Seattle Computer sold OS
    - Too many bugs
* PARC
  + Generated GUI
* Apple Lisa
  + Too expensive
  + Decided to build Macintosh so people can regularly use it
* Windows
  + Have GUI
  + Still no memory protection
  + Easier way to type
* GNU –GNU is Not Unix
  + Purpose of make sure software is free to use
  + It is really good for server

5th Generation

* Built smaller computer
* Example: Android / iOS / BlackBerry

6th Generation

* Linux is learning on Windows
* Expand to Cloud -> Amazon Web Services
  + Big change in computation
* Example: Netflix is using AWS

# Assignment #1 Note

Assignment #1 worth 10%

Assignment #2 estimated worth 20%

Assignment #3 estimated worth 20%

For assignment #1 (out of 50 marks)

Part A – Coding (30 marks)

Part B – Questions (20 marks)

Due date: 2:00pm – Friday May 20, 2016

# Hardware Review

**Hardware Architecture**



**Processor**

Fetch instruction - -

Decode instruction - - (loop)

Execute instruction - -

**Instruction Sets**

* Moves into memory
* Moves out of memory
* RISC – Risk : separate each command
  + Example: Spark, ARM
* CISC – Complex : all complexity together into smaller sets
  + Example: x86, x86\_64, AMD,

**General Registers**

* Operands
* Special purpose registers

**Program Count Register**

* Address of next instruction

**Stack Pointer**

* Next available stack frame in memory

**PSW – Process State Word**

* Condition of the CPU

**Pipeline**

1. [Fetch] [Decoding] [Executing]
2. [Fetch] [Decoding] …
3. [Fetch]…

**Modes**

* Kernel
  + Allows all instruction
* -------------------------------------Track-------------------------------------------(between kernel / user)
  + TRAP
* User
  + Subset of instruction
  + Excludes (almost or all) I/O and memory protection

**Cache**

* L1
  + Faster / expensive / smaller
* L2
  + Slower / cheaper / bigger
* Four times faster to access main memory
* Without Cache
  + Fetch 🡨-----------------------🡪 Memory
* With Cache
  + Cache Hit
    - Fetch 🡨------🡪Cache (Cache Hit)
  + Cache Miss
    - Fetch 🡨--------- Cache Miss (it takes longer time)
    - Fetch 🡨----------------------🡪 Memory

**Memory**

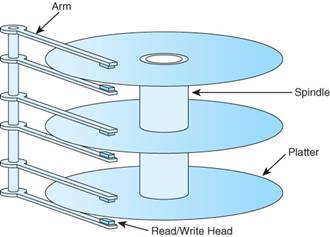
* RAM –Random Access Memory
* Read Only Memory
  + Boot loading
  + Pre wired
  + Factory set
  + Unhelpful present days

**EEPROM**

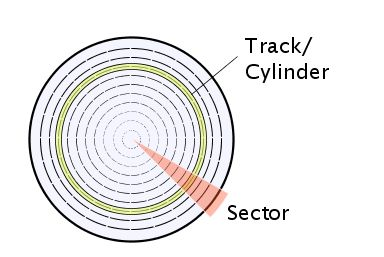
* Electrical Erasable Programmable Memory
* Make variable changeable

**Disk**

* Disks are slower, you need to wait all the way around read data from read heads



Read/Write head is not actually touching the platter, when it touches it breaks the disk



**Virtual Memory**

* Least frequently used memory
  + Map -> disk
* MMU - part of CPU
  + Maintain virtual memory