Lecture 2

This session was spent reviewing the basic components of computers and their relation to one another (as well as their relation to computer programs). Specifically, we discussed the CPU, RAM, Storage Devices, and the Clock. We also discussed the concept of a 'machine cycle'.

Machine Cycle

- The CPU executes instructions.
- A Machine Cycle is the sequence of steps that the CPU takes to execute one instruction

1. FETCH
2. DECODE
3. EXECUTE
4. WRITE BACK

- *Fetch - copy the command to the CPU*

  Decode - understand what to do with the command

  Execute - execute the command

  Write back - take results and put them back where they should be

RAM

- Random-access Memory
- Everything used by the CPU has to be loaded into RAM first:
  - Programs (set of instructions for CPU to execute). Each instruction is fetched from RAM to the CPU.
  - Data (text, images, etc)
  - etc, literally anything, the CPU does not interface with the hard drive (can't get data directly from it)
- Volatile - data is lost when the power is lost
- Slow compared to the CPU, extremely fast compared to storage devices
- Memory Stall - CPU halts to wait for data from RAM (due to speed differences)

Storage Device

- "Storage Device" is the general term for hard disks / solid state drives and so on
- Stores data (in files)
  - Program files are called executable files
- Can't fetch commands from the hard disk
- Nothing can be displayed / ran / edited / etc in hard disk, only in RAM
- Orders of magnitude slower than RAM
- Maintains state even after power-off

CPU

- Central Processing Unit
- Executes instructions as defined by its logic components / wiring
- Every CPU architecture has its own CPU instruction set , which is the set of instructions the CPU natively understands
- Only understands machine language (binary), but at some point people thought of mapping human readable 'mnemonic' commands to each machine instruction
  - As time went on, more and more was abstracted, leading to higher level assembly instructions that corresponded to multiple machine instructions (as opposed to a one-to-one relationship), and eventually modern so-called high level programming languages which completely obscure the underlying computer architecture.

Clock

- Synchronization device
- In order to keep the different computer components organized and in sync, there exists a 'clock'
- The clock releases electric pulses at a certain predefined pace (#pulses / second)
  - The clock is to the computer as a drummer is to a band
- Measured in Hertz (Hz), e.g. 2.5 gHz is 2.5 billion pulses per second.
- Each pulse of the clock ushers in one machine cycle
- This means that, in theory, a faster clock results in a faster computer.
- The clock can only be as fast as the slowest component in the computer (otherwise that component can't keep up, and all will be out of sync)

Note

Shostak asserts that a program written in assembly can outperform a modern compiler (with a high level language). Hopefully he provides some sort of example or further information next class.