

January 11, 2023

Computer Systems
Spring 2023

1 Types of Computers

- Embedded
- Centralized
- Concurrent

All computers share a common feature: execute a stored program. also the word "processor" is interchangeable with computer other types of computers

- PC
- Servers
- Super Computers

Super computers- computers that have high capability and are less available
embedded computers - which is like a network of multiple computers

When it comes to units of memory, prefer the units with powers of two.

ex gibibyte is 2^{20} or something

Personal mobile device (PMD) are phones and are battery powered

Cloud computing is done with warehouse scale computers. hardware/software interface is the connection between hard and soft.

It's made up of 2 things:

- the representation is the binary code
- something about the hardware, which reads the binary code.

In the 60's there was a memory bottleneck, that is, memory was slower than processing so computers couldn't get faster, so they focused on reducing the memory.

On the *other* hand, now we have:

- concurrent execution

- hierarchy of memory - which may benefit or slow down computers
- energy efficiency - basically the only problem now is energy and other holistic problems with computers
- memory leaks that cause cybersecurity problems

Performance is a key focus of the class, key things that help performance:

1. algorithms - which control the # of operations executed.
2. programming language
3. compilers
4. architecture?
5. processor
6. memory
7. I/O system performance impacts the computer's performance but this is covered in the **Operating Systems class**.

These will be covered throughout the semester and seem important to understand. Here's how to make code faster:

- data level parallelism
- instruction level parallelism
- memory hierarchy optimization
- thread level parallelism

2 The Great Eight Ideas

1. Design for Moore's Law. ...
2. Use Abstraction to Simplify Design. ...
3. Make the common case fast
4. Performance via parallelism

5. Performance via pipelining
6. Performance via prediction
7. Hierarchy of memories
8. Dependability vs redundancy