**DSC-204A: Scalable Data Systems, Winter 2024**

# 2: Basics: Computer Organization, Operating systems, Storage

*Lecturer: Hao Zhang* *Scribe: Junda Chen, Yiyang Yu*

## 1 Class Roadmap: History of Compute and Data.

It's important to build a global picture of the history of compute and data. Another thread to understand this is asking: "Which is the most valuable company in tech".

- 1980-2000: Personal computer. Companies include Intel, Microsoft, Sun, IBM, Apple (still dominating the laptop market). Companies like intel, Microsoft, sun, IBM, and Apple emerged. Intel made the CPU chips, and Microsoft constructed the operating system that we use. Apple developed the first personal computer, and now they are dominating the laptop market. IBM built super computer.

- 2000-2016: Cloud computing. The dot-com bubble and the rise of network technology make people think of putting computer together instead of building a mainframe. At the start of 2000, there was a dot-com bubble where everyone starts doing things related to network. Cisco once became the most valuable company in the world. People started considering putting more computers together as clusters, which is cheaper than building a super computer but with the same amount of compute power. Cloud was designed as computers put together that are used by others who need compute and store data. The prominent companies during this era include Microsoft, aws, and Cisco. Amazon was remodeled from a bookstore to the largest global company now.

- 2008-2020: Big Data. Once personal computer prevail, people start to upload data to cloud, and company start to build recommendation systems etc to provide better service and many more using these big data. This is the era of big data mining. Companies include Meta, Google, Netflix. People developed analytical tools to study data.

- 2016-Now: Machine Learning Systems. Google DeepMind, NVIDIA, OpenAI. We want to build intelligence using data. Google stands out during this era with their research papers and products based on their AI. OpenAI shippied ChatGPT. NVIDIA also makes a lot of money by producing chips.

## 2 Foundation of Data Systems

This class focuses on two big parts:

- Computer Organization: How computer is organized (logically). This includes (1) understanding how data is represented, (2) understanding waht are processors, memory, storage

- Operating System Basics: (1) Processes (scheduling), (2) file system, (3) memory management.

## 2.1 Computer Organization: What is a computer?

Peter Naur says: "A programmable electronic device that can *store*, *retrieve*, and *process* digital data." This definition emphasizes the importance of data as well as computer interactions with data. However, this definition is too abstract.

### 2.1.1 Basics of Computer Organization

- Hardware: The electronic machinery (wires, circuits, transistors, capacitors, devices, etc.)
- Software: Programs (instructions) and data. Actually, program is also a type of data stored in a computer that is used to instruct the device how to process other forms of data.

Compute and storage is the most important parts we care in the class.

- To store and retrieve data, we need: disks, memory (why do we need both? - disks is persistent, memory is ephermal data) Computer has a storage hierarchy, and there are fast and slow storage.
- To process data: processors (CPU and GPU)
  - At the era when Moore's Law still works, we just produce CPU with higher IPS each year.
  - GPU is now the focus now that Moore's law died.
- To retrieve data from remote: Networks. Networks allow forming clusters between computers.

### 2.1.2 Hadware: the key parts

- **Processor** (CPU, GPU, etc.): on the motherboard. Hardware that orchestrates and executes instructions to manipulate data as specified by a program.
- **Main memory** (aka DRAM): hardware to store data and programs that allows very fast location / retrieval; byte-level addressing scheme. It is in byte-level, which is the smallest unit that CPU and GPU can address.
- **Disk** (aka secondary / persistent storage): similar to memory but persistent. If you turn off the computer, the data stored on disk is saved. Slower, higher capacity / cost ratio (cheaper). Has various addressing schemes (sequential access will be faster than random access).
- Network interface controller (NIC): hardware to send / retrieve data over network of interconnected computers/devices. Network allows scale up, forming bigger clusters and allowing more computes.

In reality, the personal computer is much more complicated. Inside a computer, you would have: motherboard, SATA connectors, integrated Ethernet chip, PCI express (PCI-E) slots to put in your GPU, CPU socket, DDR2 memory socket, etc. CPU is installed in socket. SATA connectors can connect multiple disks. Slots called PCI-E is where GPU is plugged in.

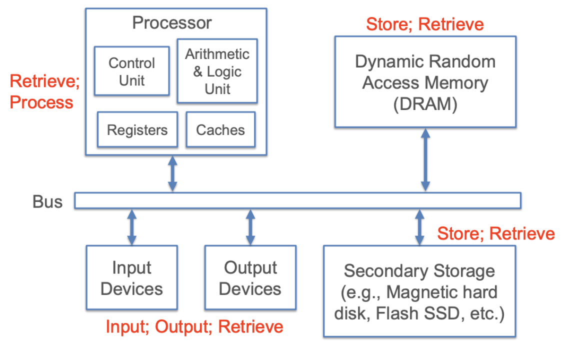


Figure 1: Very classical figure to illustrate how the devices are connected to each other. Memory and secondary storage are used to store and retrieve data. Processor manages data retrieve and process.

### 2.1.3 Key Parts of computer software:

Users interact with computer by interacting with software.

- **Instruction**: Instructions are non-human-readable. Depend on instruction set architecture (ISA). A command understood by hardware; finite vocabulary for a processor. Bridge between hardware and software.
- **Program (code)**: A collection of instructions for hardware to execute. Higher-level language (compared to instruction) that is human readable. Can be compiled into instructions.
- **Programming Language (PL)**: Human readable formal language to write programs; at a much higher level of abstraction than ISA. *Maybe there is a trend to let LLM deisgn PL and write programs using the PL it generated. People has been debating for decades which PL is better.*
- **Application Programming Interface (API)**: A set of functions (interface) exposed by a program / set of programs for use by huamn / other programs. e.g.: `getdate()` is a function that abstracts the getting of date. APi is usualy defined by some software vendors for developers to do programming.
- **Data**: Digital representation of information that is stored, processed, displayed, retrieved, by a program.

### 2.1.4 Main Kinds of Software

- **Firmware**: Read-only program "baked into" a device to offer basic hardware control functionalities. For example: Hardware driver.
- **Operating System (OS)**: Collection of interrelated programs that work as an intermediary platform / service to enable application software to use hardware more effectively. It orchestrates all the resources available, like CPU and GPU. Examples include Linux, Windows, and MacOS.
- **Application Software (i.e. App)**: A program or a collection of programs to make the life of a user. Examples include Excel, Chrome, and PostgreSQL.

*Quick Question: What is the application look like in a cluster? How do we run an application that spans cluster-wide?*

Answer: Similar to single machine application, we install an application (or a part of application) on each node, then try to connect these parts together.

## 2.2 Digital Representation of Data

### 2.2.1 Basic

- **Bits**: The smallest unit of data representation. All digital data are sequences of 0 & 1 (binary digits). Why do we use bits? It is because of Physics. High-low / On-off electromagnetism on disk is easy to store and represent.
- **Data type**: First layer of abstraction. Interpret a bit sequence with a human-understandable category of information; interpretation is fixed by the PL. e.g.: Boolean, Byte, Integer, Float (floating point), Character, String
- **Data Structure**: The second layer of abstraction on top of data type. It organize multiple instance of same varied data types as a more complex object with specified properties. e.g.: Tuple, Graph, list, int

*Quick Question: What can replace bit?*

Answer: Quantum bits! In quantum computing, we represent data using "qbit".

### 2.2.2 Count everything in binary

- Because computer store things in binary, computer use Base 2 to represent number.
- e.g. Represent $15213_2 = 0011\ 1011\ 0110\ 1101_2$
- How to represent negative numbers? *(leave it for next class...)*

### 2.2.3 Encoding Byte Values

**Bits and Bytes.**

- 1 Byte = 8 bit
- Why is "1 Byte = 8 bit"? Just a historical engineering decision to group bits.
- Why is it a power of 2? Practicality and standardization.
- Byte as the smallest addressable unit. Computer cannot address things smaller than a byte.

**Bytes → Data types: bool, int, float, string, ...**

- The size and interpretation of a data type depends the PL
- Boolean: Represent Y/N (true / false). Just 1 bit needed, but actual size is almost always 1B (meaning we use 8 bit to include a 1 bit information → 7 bit wasted!)

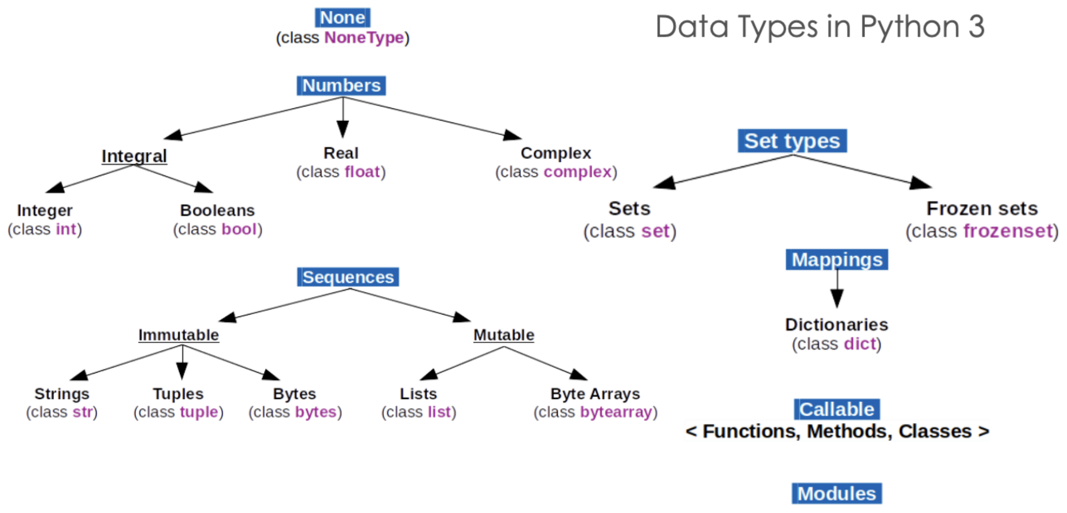


Figure 2: Digital Representation of data in Python 3

- Integer: typically 4 bytes (32 bits), but also have some variants (short, unsigned, etc.). Java int can represent $-2^{31}$ to $(2^{31}-1)$. C unsigned integer can represent 0 to $(2^{31}-1)$

**Quick Questions 1**

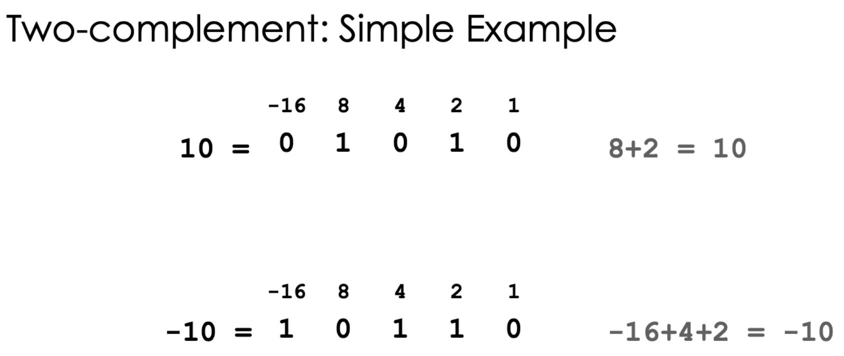
1. How many unique items can be represented in 3 bytes? - $2^{24}$
2. How many bits are needed to distinguish 97 data items? - 7 bits $= ceil(log_2(97))$
3. How to convert from decimal to binary representation? - Division by $2^k$ and put 1 at bit position k; filling remaining positions with 0s.
4. How to convert binary to decimal? - Summing the power of 2 at position k

**Quick Questions 2: What is wrong about this function?**

```
void show_squares() {
    int x;
    for (x = 5; x <= 5000000; x *= 10)
        printf("x = %d, x^2 = %d\n", x, x * x);
}
```

Program Output:

```
x = 5 x^2 = 25
x = 50 x^2 = 2500
x = 500 x^2 = 250000
x = 5000 x^2 = 25000000
x = 50000 x^2 = -1794967296
x = 500000 x^2 = 891896832
x = 5000000 x^2 = -1004630016
```



Two-complement: Simple Example

```
       -16  8   4   2   1
10 =    0   1   0   1   0      8+2 = 10

       -16  8   4   2   1
-10 =   1   0   1   1   0      -16+4+2 = -10
```

Figure 3: Simple example about two's complement

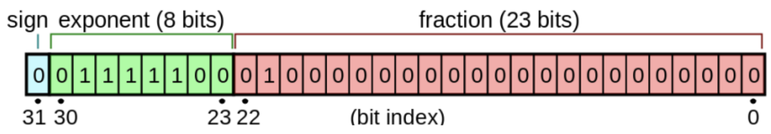


Figure 4: the float IEEE standard - binary32

Answer: Overflow.

### 2.2.4 How computer represent integers

- Two-complement.
- Caveat: the highest bit is a ”value”, not just the ”sign”!
- Positive and negative numbers are ”complement” in binary.

### 2.2.5 Floating Points Data Types

Float: IEEE-754 single-precision format is 4B long; double-precision is 8B.

$$(-1)^{sign} \times 2^{exponent-127} \times (1 + \sum_{i=1}^{23} b_{23-i} 2^{-i})$$

$$(-1)^0 \times 2^{124-127} \times (1 + 1 \cdot 2^{-2}) = (1/8) \times (1 + (1/4)) = 0.15625$$

Figure 5: The formula to do the actual calculation

- Java and C `float` data type is single precision (32 bits); Python's `float` is double precision (64 bits).
- In the standard float32 format: sign(1 bit), exponent (8 bit), fraction (23 bit)
- Exponent: The offset of the number. This represents how large the range this data type can represent.
- Fraction: The precision of the number. Can be interpreted as the significant bits of the number.

There are more float standards: double (float64, 8B), half (float16, 2B). They come with different number of bits for exponent and fraction. **FP16** is now common for deep learning parameters, probably one of the most important float data type. It has native support in PyTorch, TensorFlow, etc. API also exist for weight quantization.

Which float format is "better"? That is an open question.

### 2.2.6 FP16 vs FP32

NVIDIA shipped A100 (Ampere) with bf16 in this arch. It shows that bf16 is better than fp16 in training model. Using FP16 give better FLOPS than FP32.

**FLOPS**: floating point operations per second **Mix-precision training**: using automatic mixed precision for major deep learning frameworks.

## 3 Practice Questions for next class

- How many space do I need to store GPT3
- What does exponent and fraction control in float point representation?
- What is the difference between bf16 and fp16?