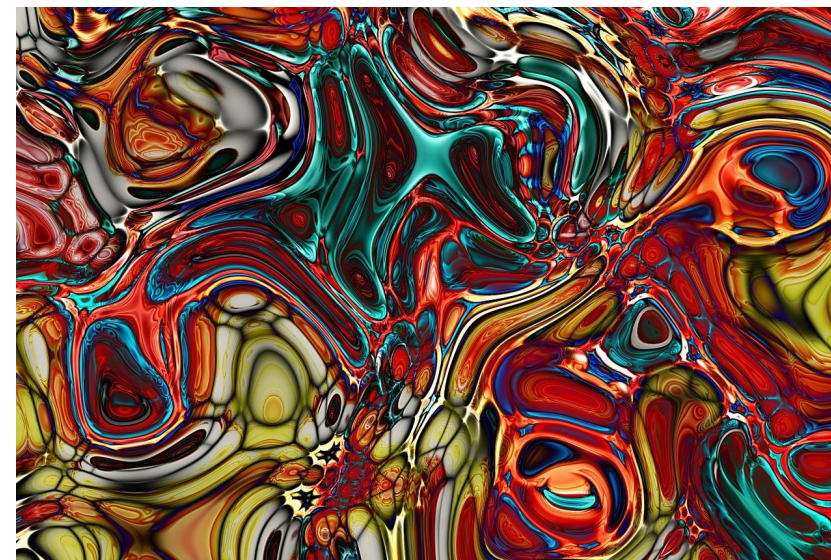
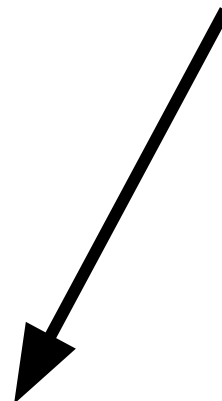
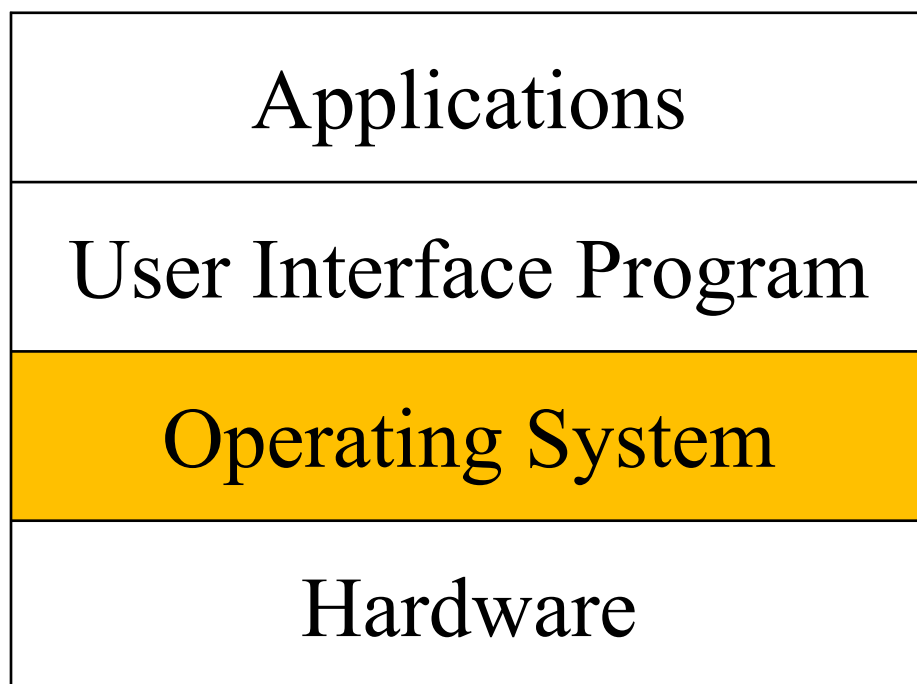


# System Software

# Operating System



# Where does the OS fit?





# OS Zoo



# OS Zoo

## What will the desktop, server, mobile OS optimized to do?

### Desktop and Server



Mac OS X



### Mobile OS



# OS Functions

$$F = G \frac{m_1 m_2}{r^2}$$

$$\phi(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

$$E = mc^2$$

$$ds \geq 0$$

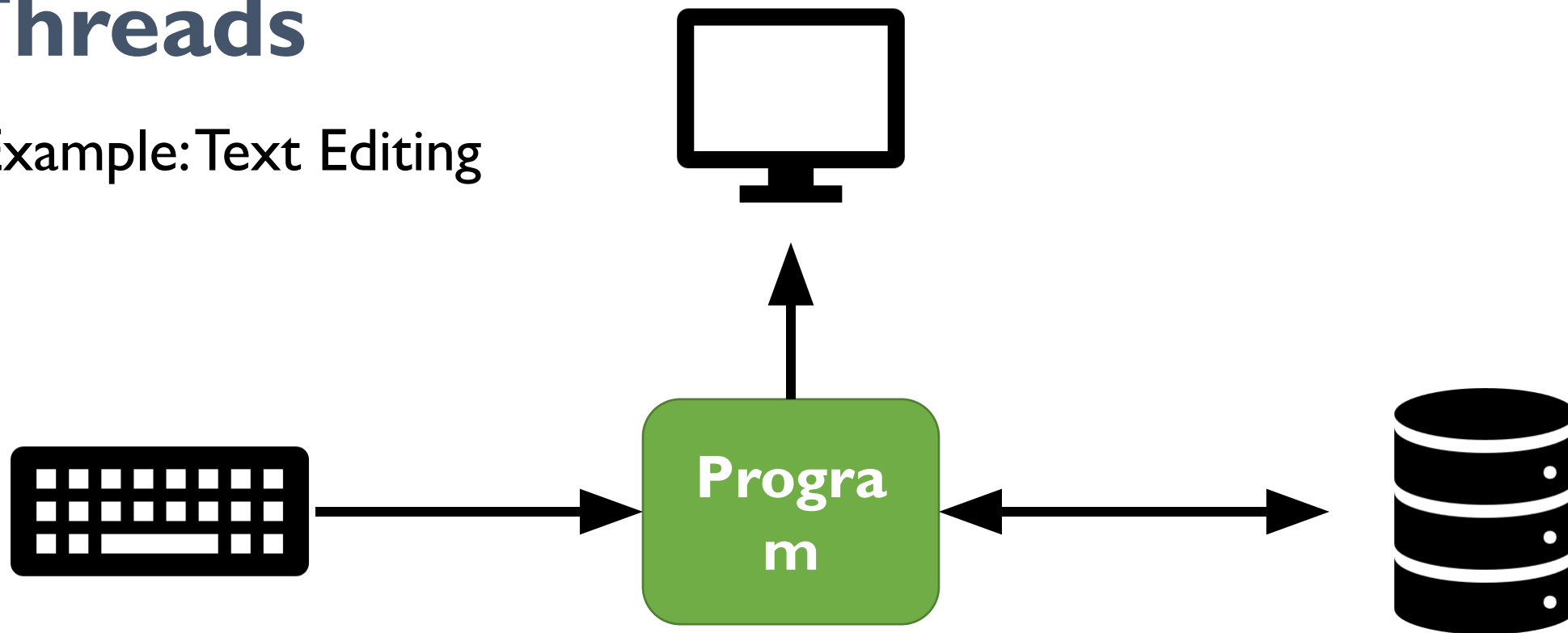
$$\frac{df}{dt} = \lim_{h \rightarrow 0} \frac{f(t+h) - f(t)}{h}$$

# OS Functions

- Process Management
- Memory Management
- I/O Management
- File System
- Security

# Threads

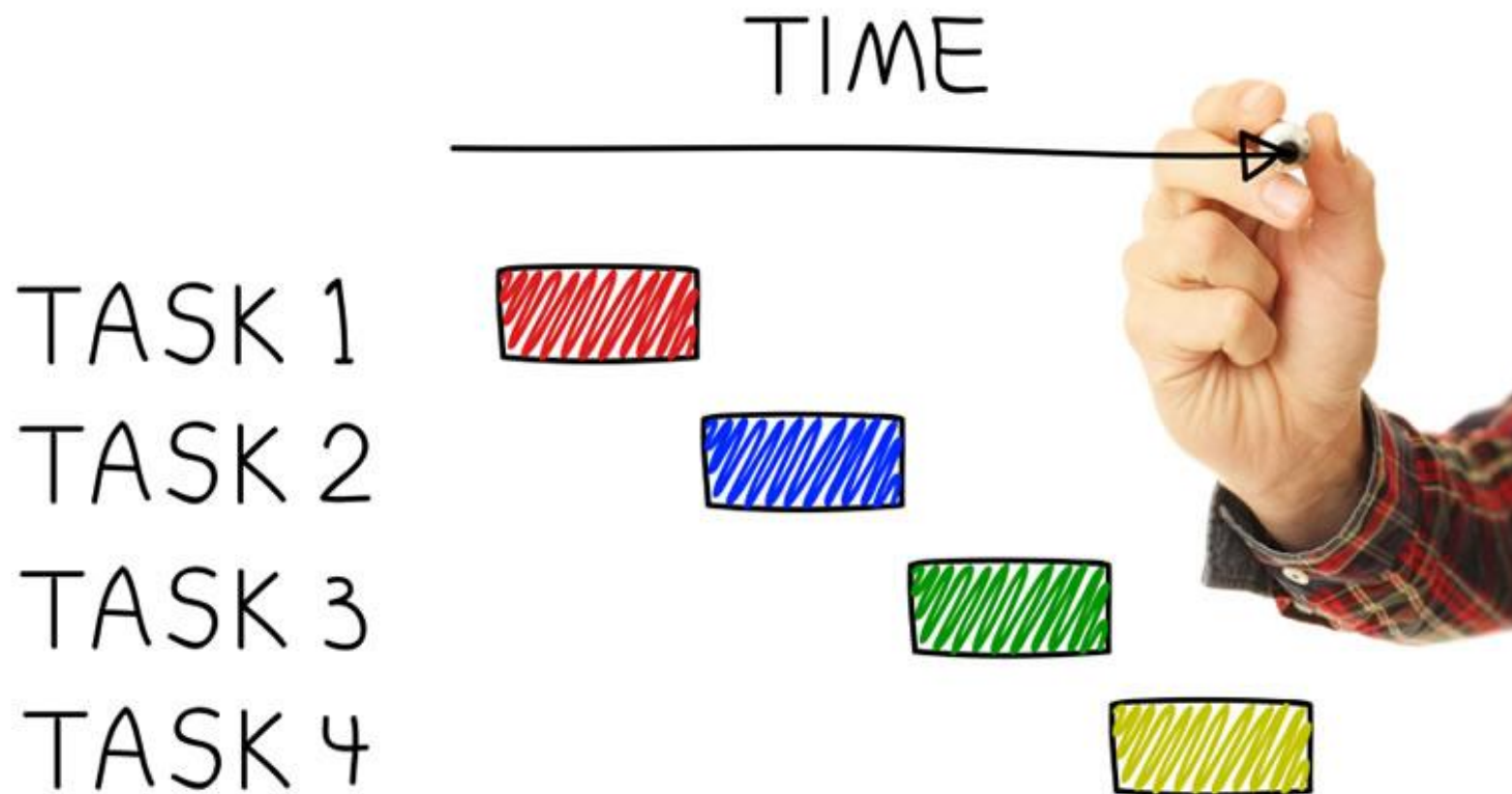
Example: Text Editing





# Scheduler

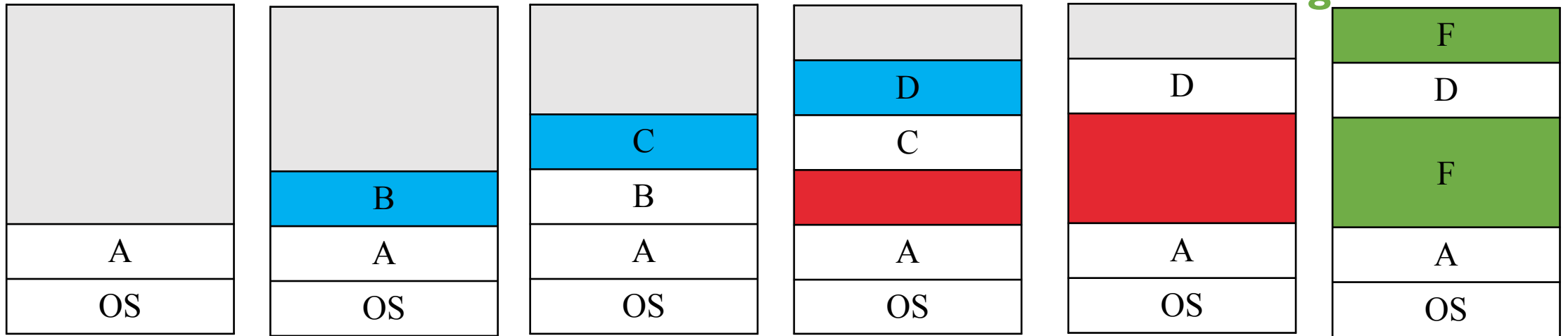
**Determines the priority in schedule each task, process, or thread**



# Memory Management

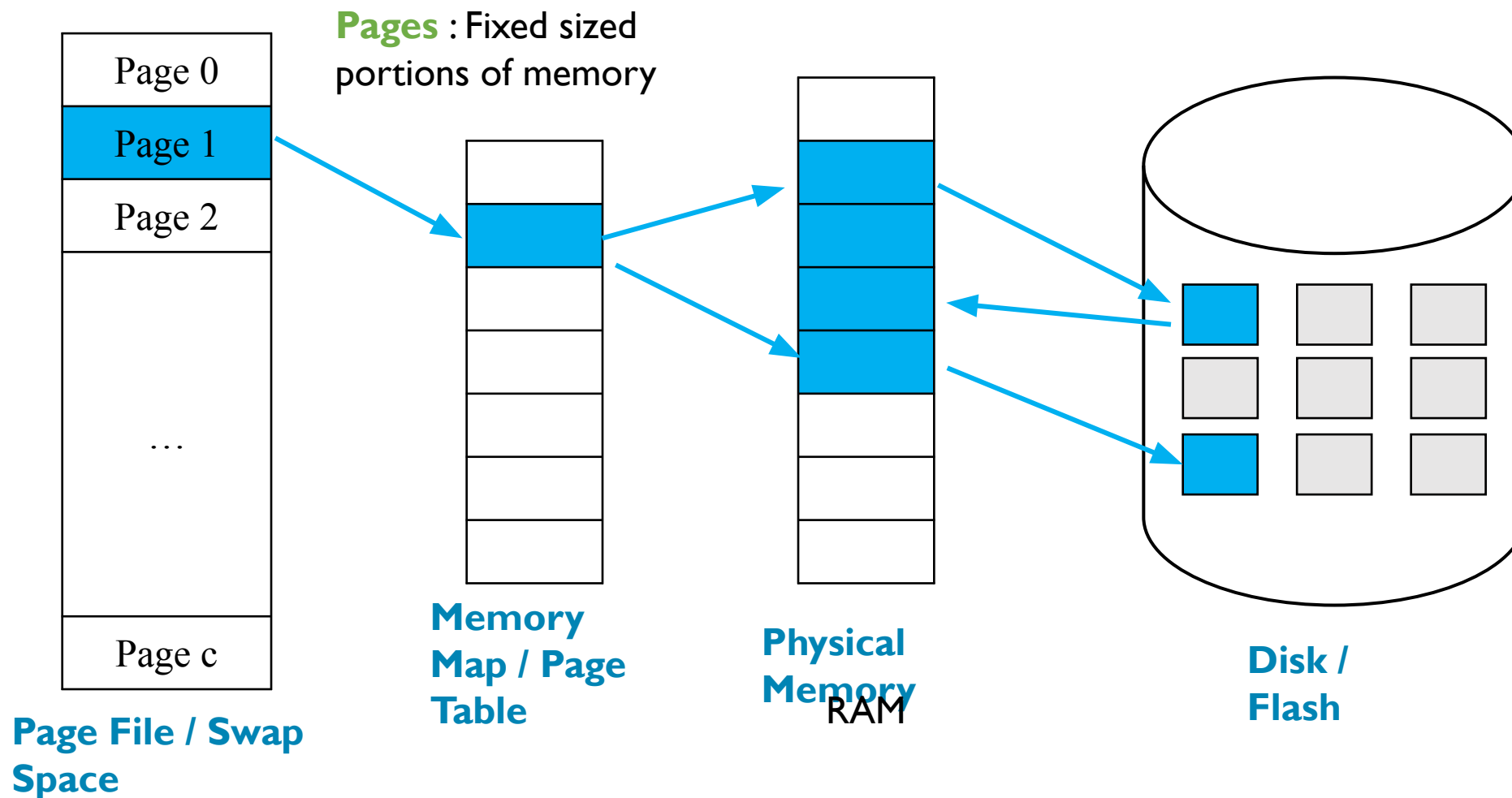
- Limited Memory
- Data can be deleted
- Data can be written anywhere

## Data Fragmentation



# Virtual Memory

Allows users to write programs for an extremely large virtual address space that can't fit in the RAM





# I/O Management

Applicati  
on



Resource Allocation Layer

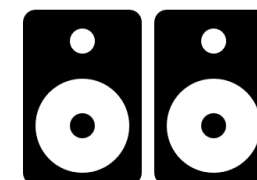
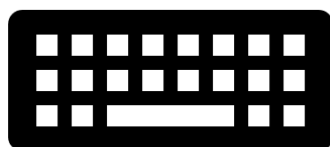
Screen Device  
Driver

Keyboard  
Device Driver

Mouse Device  
Driver

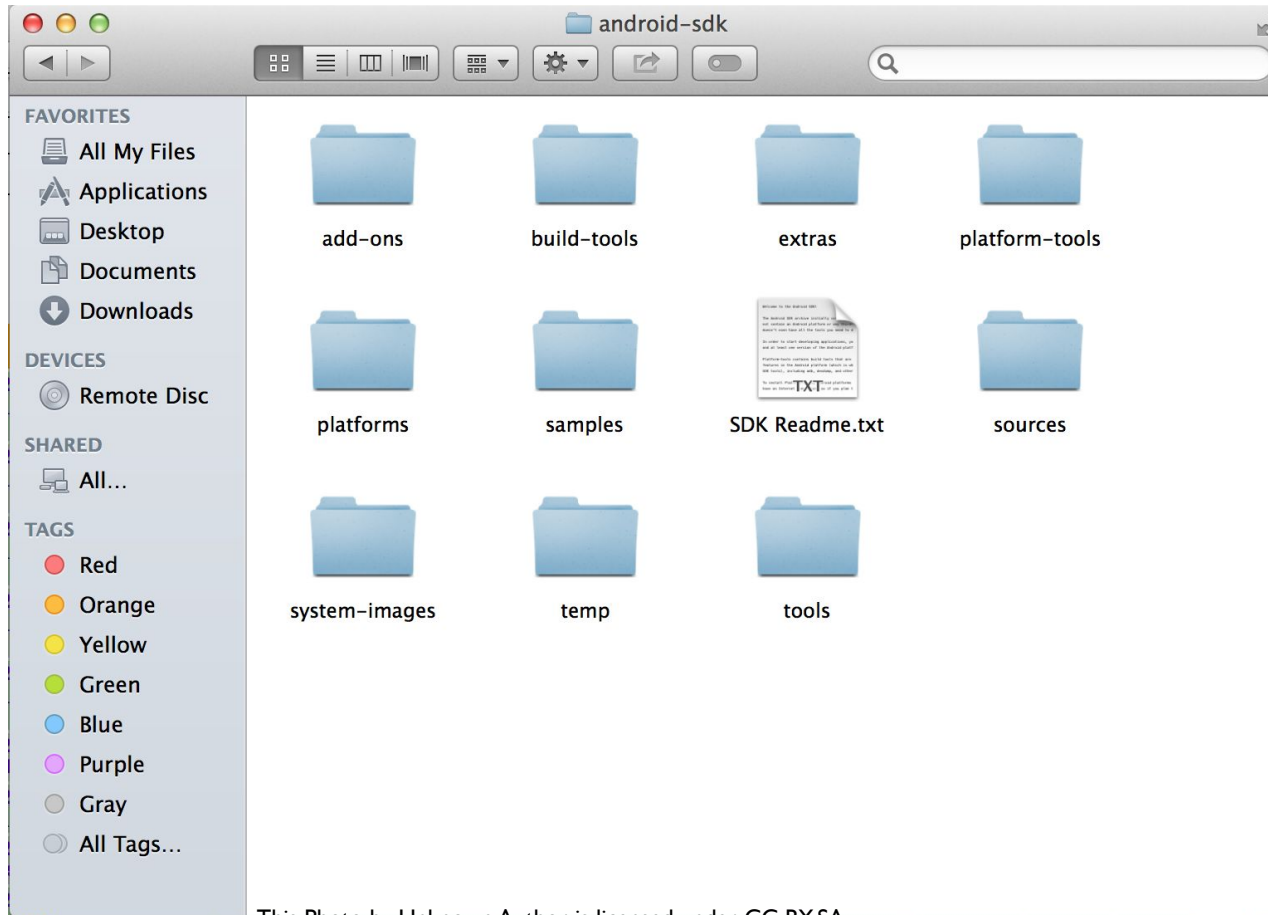
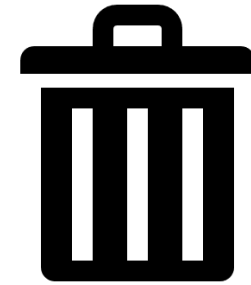
Hard Drive  
Device Driver

Sound Generator  
Device Driver



**Device driver** : allows the OS to control a device

# File System



This Photo by Unknown Author is licensed under [CC BY-SA](#)

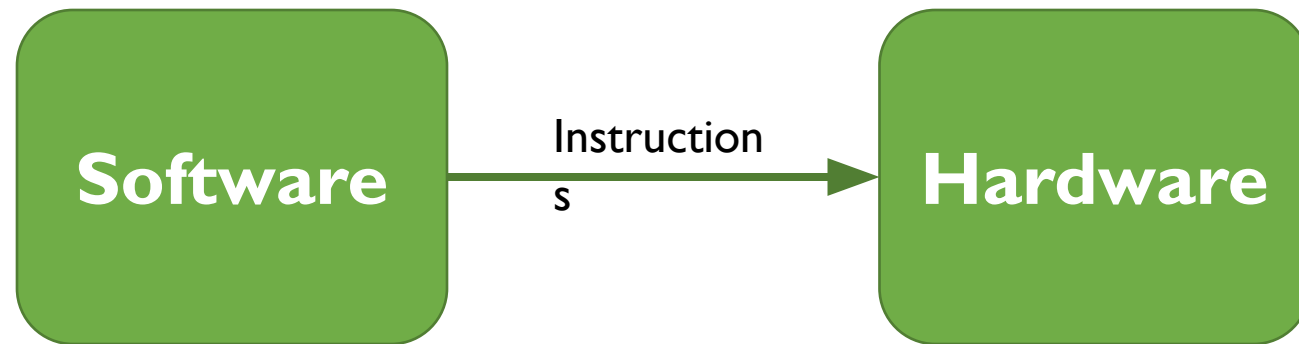
A detailed architectural drawing of a building plan is the background. It shows various rooms, corridors, and structural elements. Overlaid on the drawing are several tools: a pair of compasses on the left, two markers (orange and green) at the top, a large transparent ruler on the right, and a black pen with its cap off in the lower right. The text 'Instruction Set Architecture' is written in a bold, dark blue font on a white rectangular background in the lower-left quadrant.

# Instruction Set Architecture



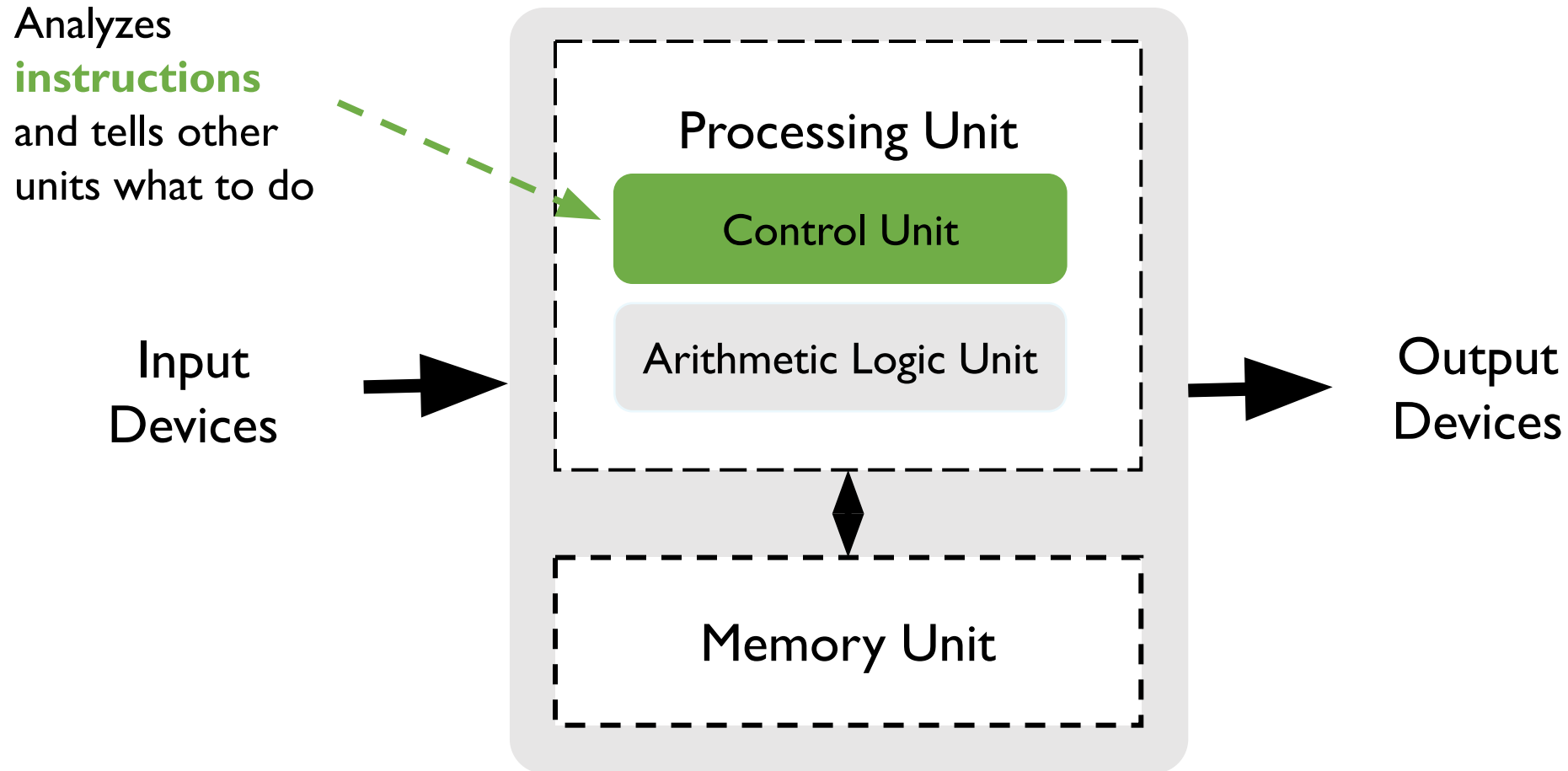
# Instruction Set Architecture (ISA)

- Defines how the processor is controlled by the software



- Learning ISA helps you to run more efficient code and understand compiler output

# Von Neumann Architecture

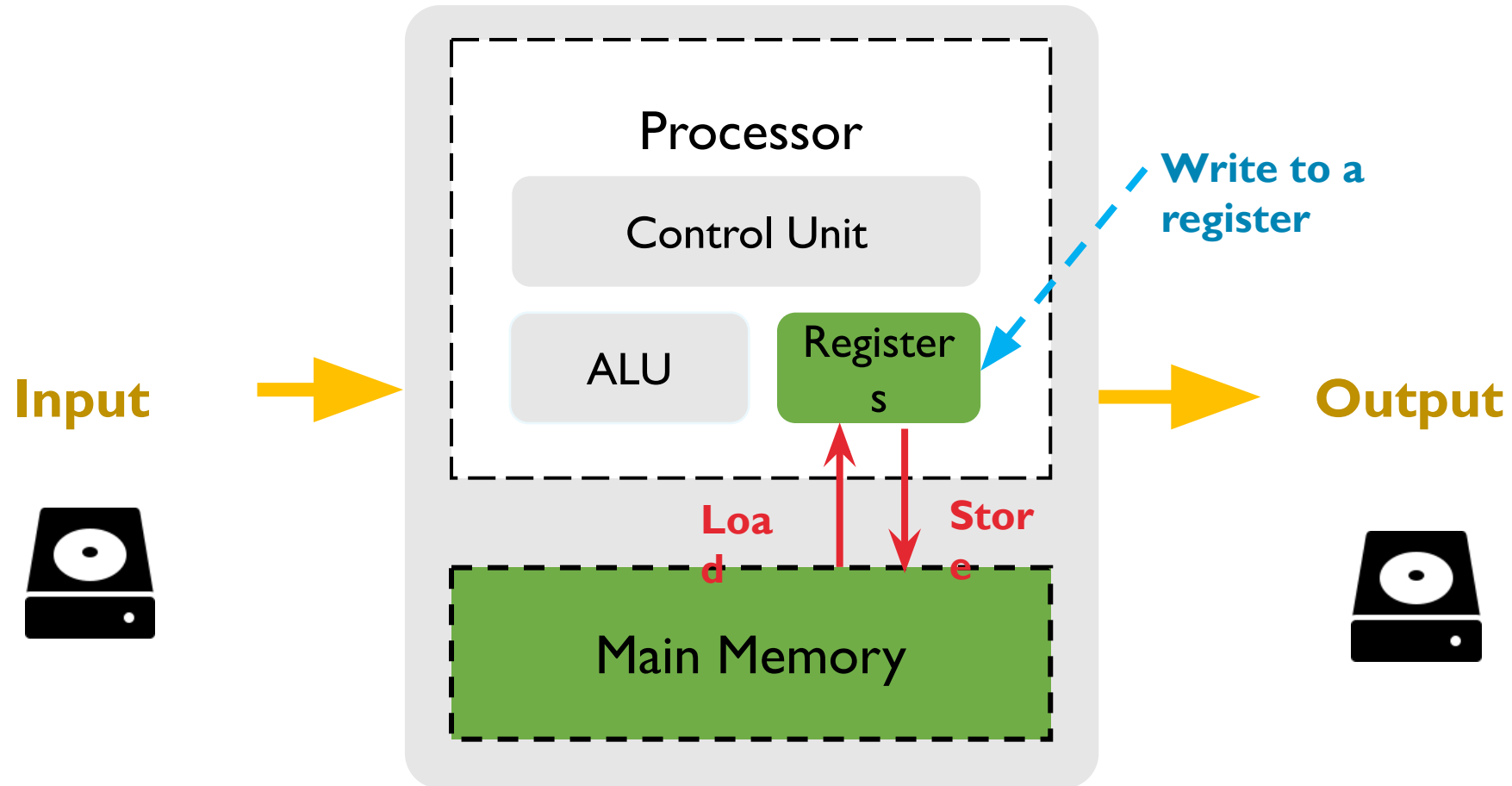


# Instruction Set

- Memory Operations
- Arithmetic and Logic Operations
- Control Flow Operations
- Co-processor Instructions



# Memory Operations



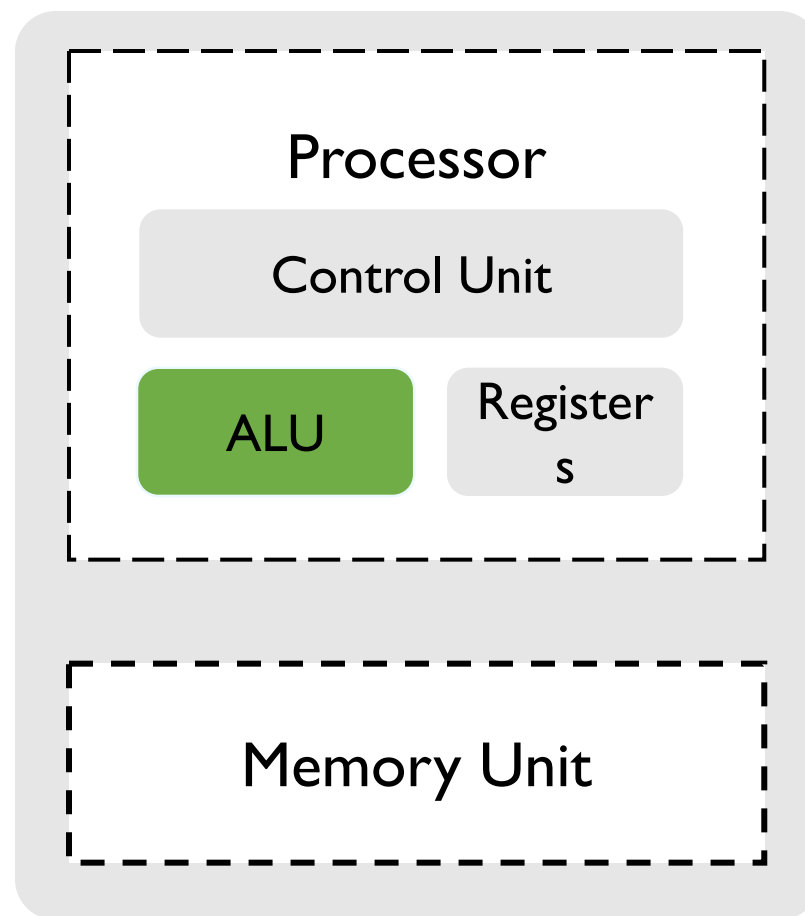
# Arithmetic and Logic Operations

## Arithmetic Operations

**+ - X**  
**/**

## Conditional Operations

**>**  
**<**

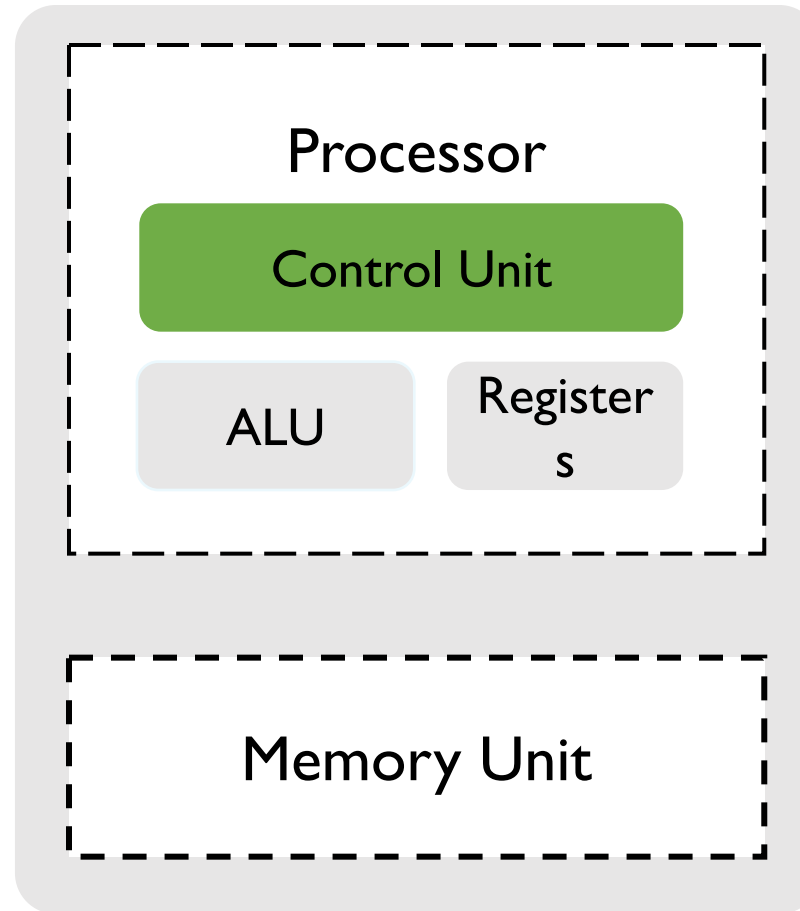


## Logic Operations

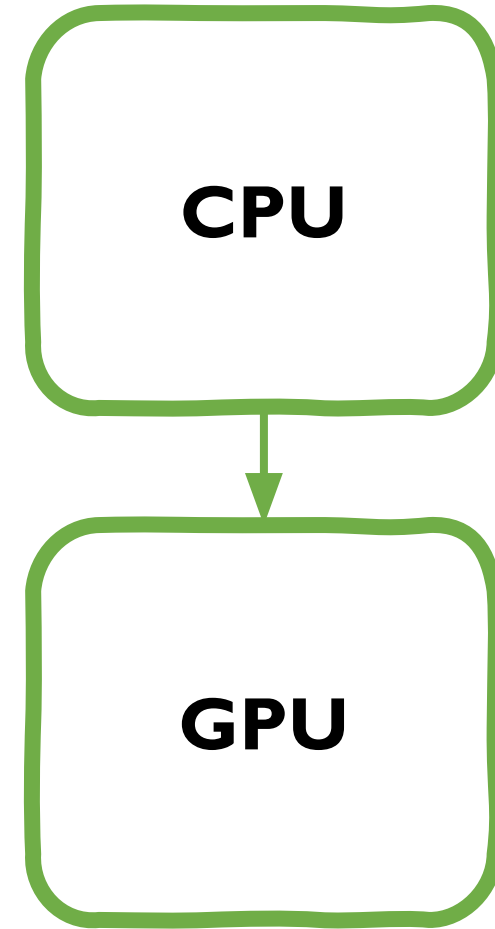
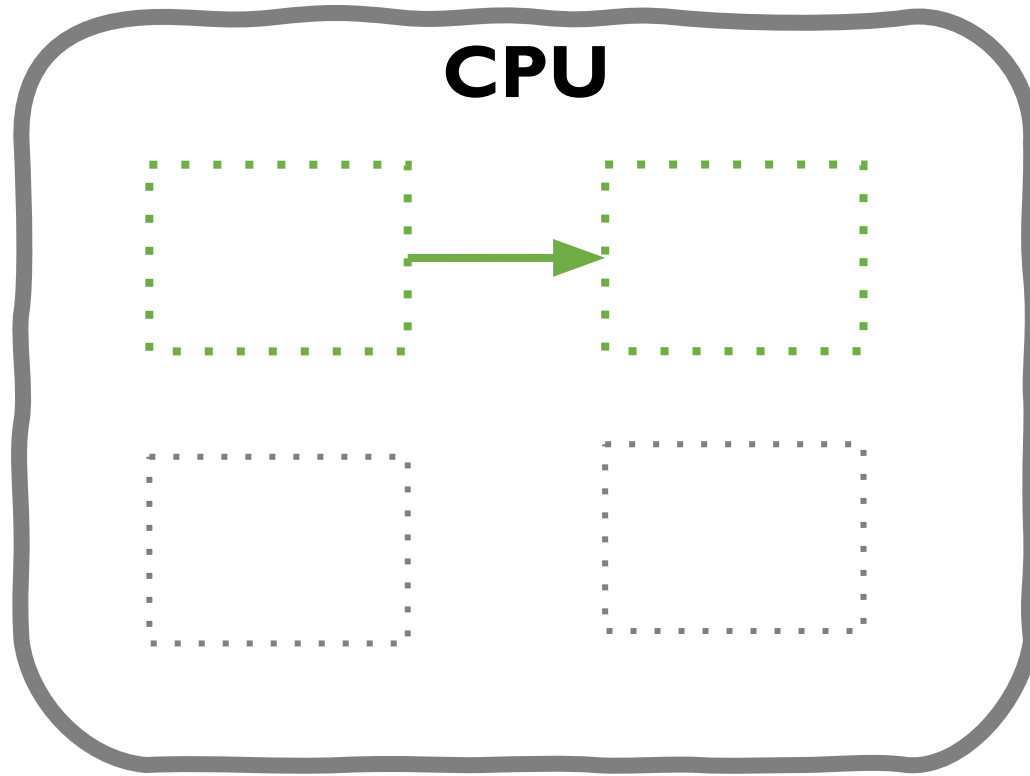
**AND**  
**OR**  
**NOT**  
**XNOR**

# Control Flow Operations

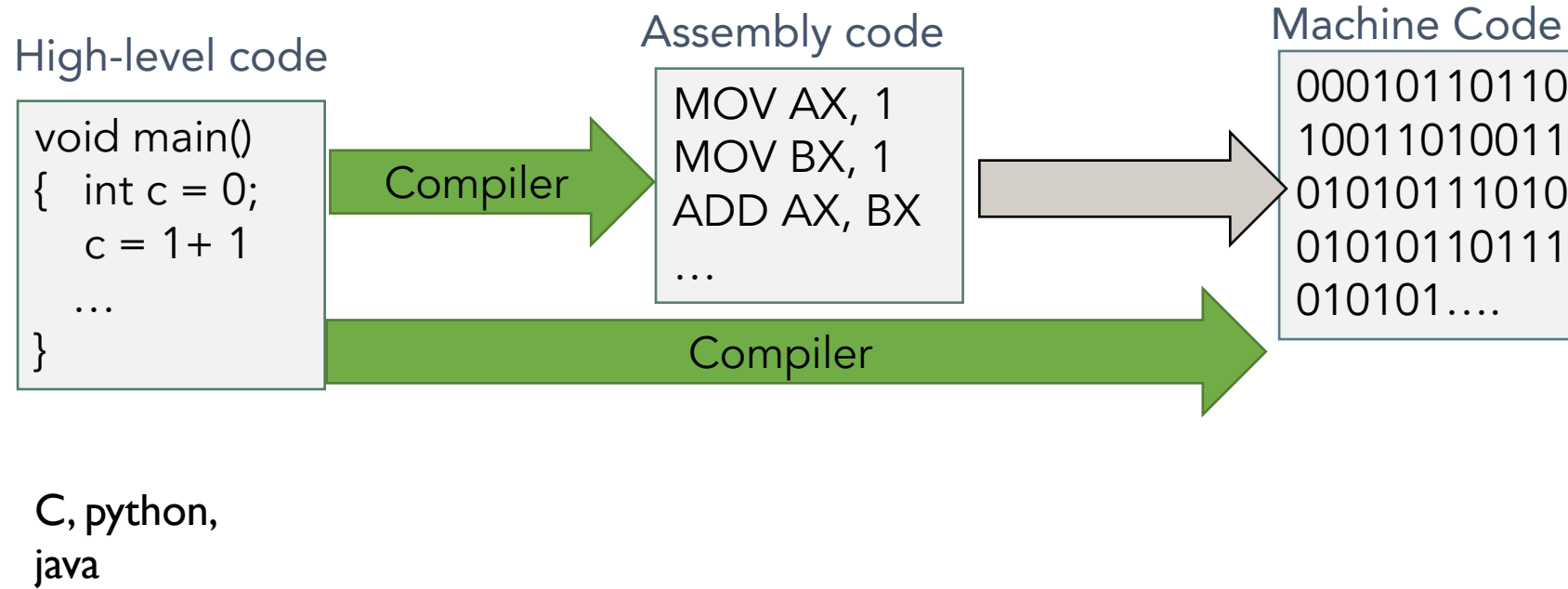
- IF ... ELSE ...
- Loops
- Function Calls



# Co-processor instructions

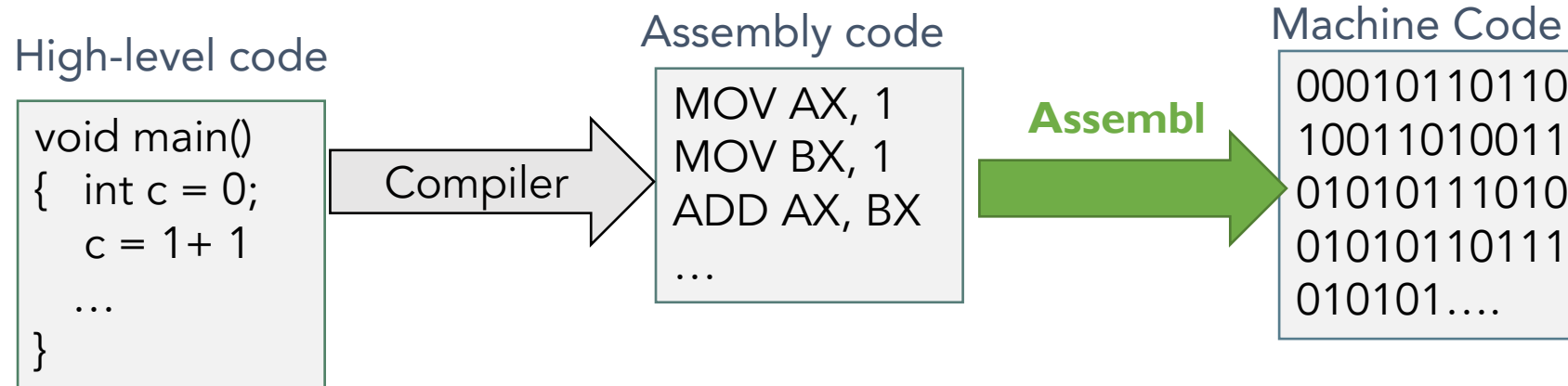


# Compilers





# Assembler



# Interpreter

High-level code

```
void main()
{  int c = 0;
   c = 1+ 1
   ...
}
```

Interpreter

Interprets code one line at a time

Machine Code

```
00010110110
1
00110100110
1
01011101001
0
10110111010
1
01....
```