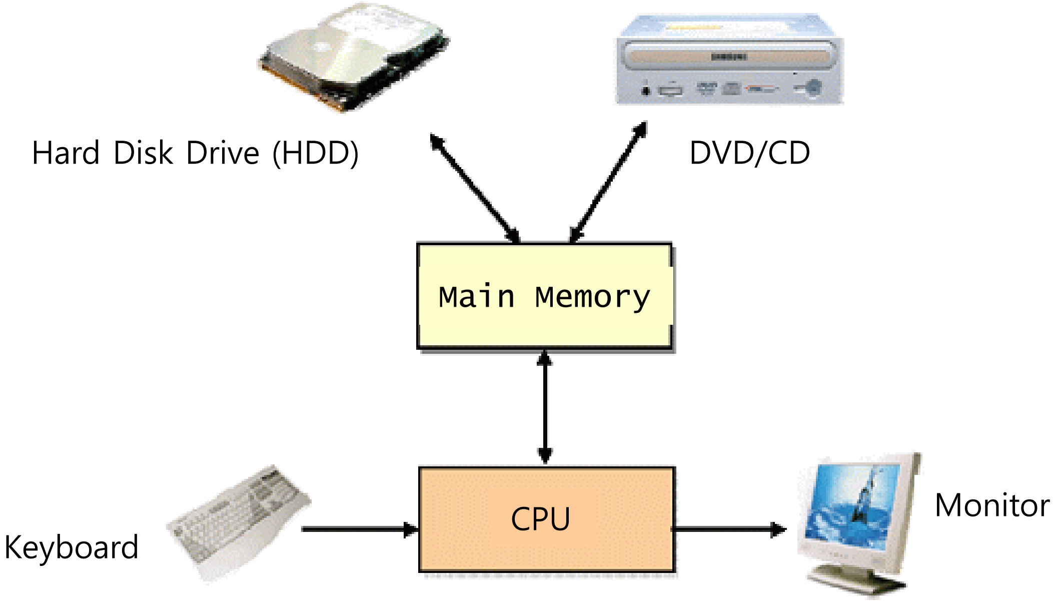
Lecture 2 : Computer System and Programming

Computer?

- a programmable machine that
  - Receives input
  - Stores and manipulates data
  - Provides output in a useful format

Computer System

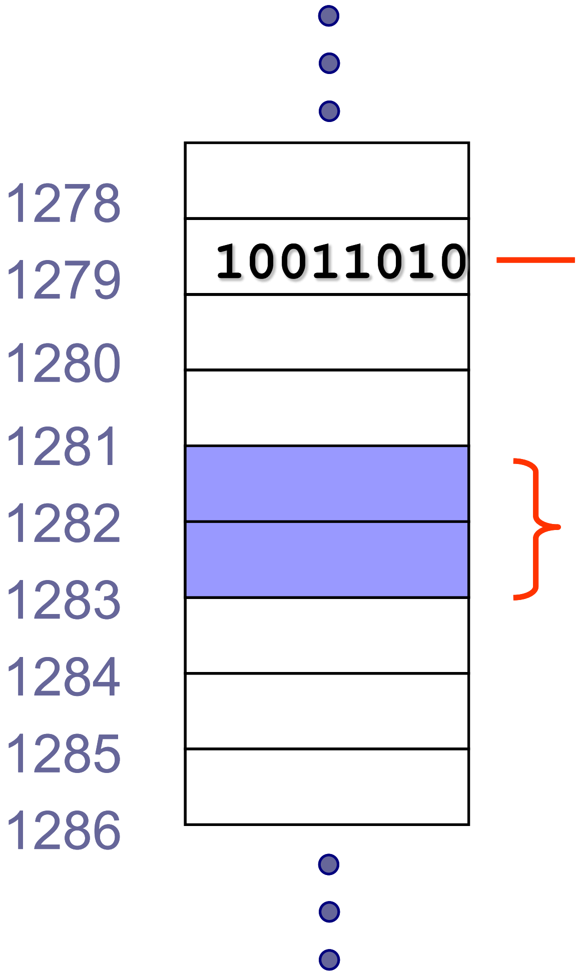
- **Computer System**
  - □ Hardware + Software
- Computer Hardware



Computer Hardware

- **CPU (Central Processing Unit)**
  - Processing program instructions (one by one)
  - Basic program instructions : add/subtract/multiply/div, read/write, jump, test
  - **Cache** : duplicating original data stored in slow storage into faster storage
- **Main Memory (e.g. RAM)**
  - **Volatile** : when power turned off, data in the memory will be erased
  - Storing progam and data
  - Fast, small, and expensive
- **Secondary Memory (e.g. HDD, CD/DVD, …)**
  - **Non volatile**
  - Relatively slow, large, and cheap
- **I/O(Input/Output) Device**
  - Help interaction between computer and human beings.
  - Keyboard, mouse, monitor, etc

Memory and Data

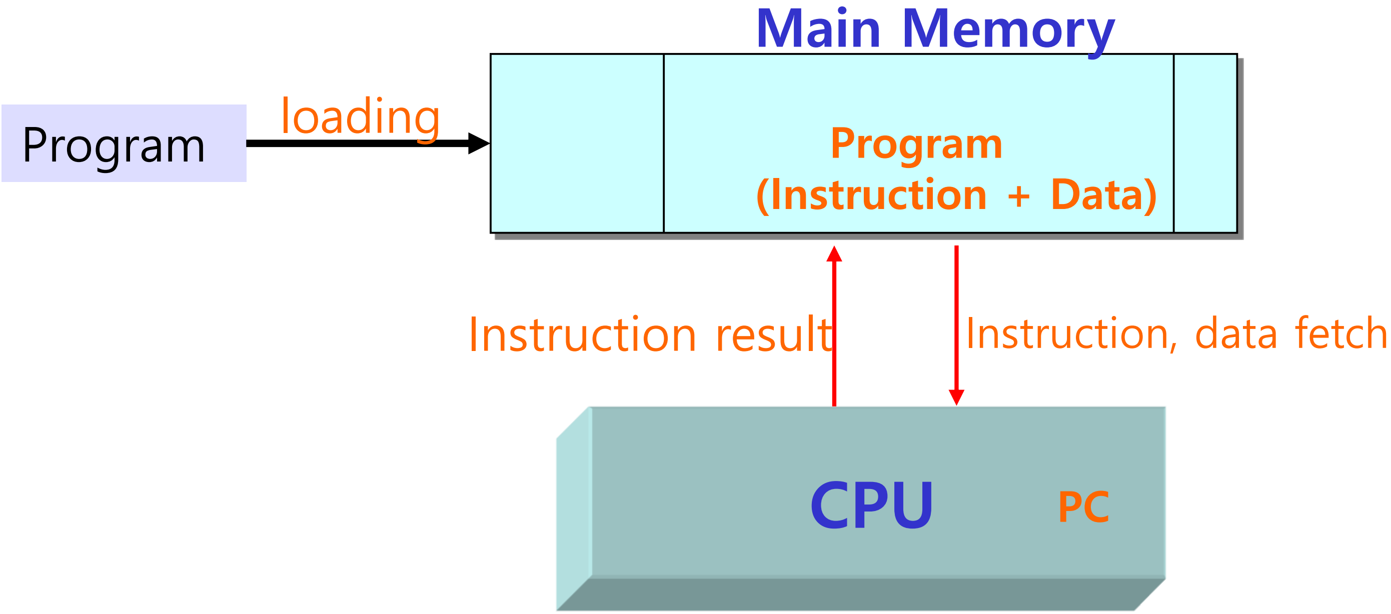


Each memory cell stores
1 byte data

Bigger data values are
Stored in consecutive memory cells

Program Execution

- **von Neumann architecture**

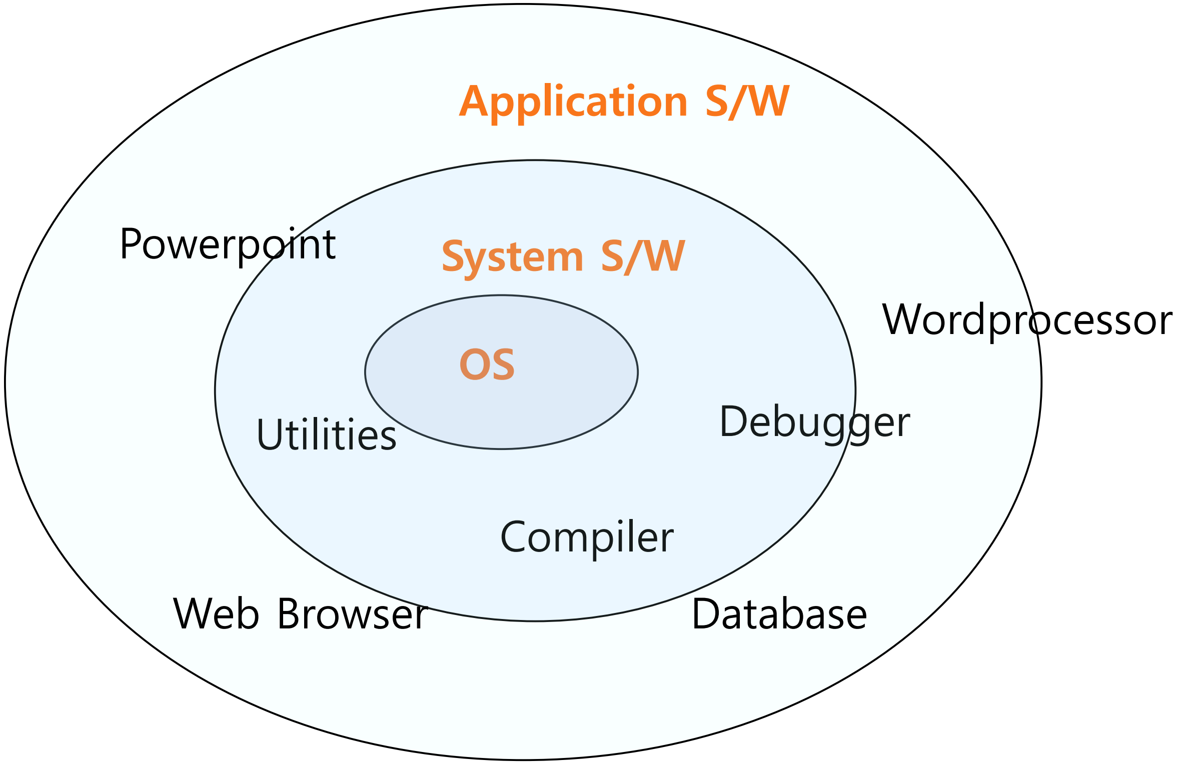


Software

- system software
  - Efficient management of computer system and resources
  - Operating System, compiler, debugger

- application software
  - All kinds of software other than system software
  - Wordprocessor, spreadsheet(excel), graphics SW, artificial intelligence SW, Game SW, Statistics SW, medical SW

Software Layers



Data Representation

- Binary number
  - □ Computer uses binary number
  - □ 1bit can represent 0 or 1
  - □ N bit number can represent up to 2^N

1 bit	2 bit	3 bit	4 bit
0	00	000	0000
1	01	001	0001
	10	010	0010
	11	011	0011
		100	0100
		101	0101
		110	0110
		111	0111
			1000
			1001
			1010
			1011
			1100
			1101
			1110
			1111

Binary number, decimal number

- Decimal number
  - Use 0 - 9
  - 182 $= 1 \times 10^2 + 8 \times 10^1 + 2 \times 10^0$

    $= 1 \times 100 + 8 \times 10 + 2 \times 1$
- Binary number
  - Use 0 and 1
  - 1101_2 $= 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0$

    $= 1 \times 8 + 1 \times 4 + 0 \times 2 + 1 \times 1$

    $= 13$

Programming Language

- Language for programming computer processing
  - Machine readable language designed to express computations that can be performed by a computer
  - Specify behavior of machine, express algorithms
  - Human-Computer Communications

- Machine language
  - Binary code

```
1001   0001   store value at address 0001 into accumulator
1100   0010   add value at address 0010 into accumulator
```

- Assembly Language
  - Symbolization of machine language binary code

```
LOAD   Y
ADD    Z
```

Programming Language

- High level language
  - Easy to use (read and write) , human friendly
  - Programmer does not need to know details of machine control.
  - More portable (machine independent)

    X = Y + Z

- example
  - FORTRAN, COBOL, BASIC, C, C++, Java

High Level Programming Language

- FORTRAN(FORmula TRANslation)
  - Created in 1957 by John Backus (IBM)
  - General purpose PL especially suited For scientific computation

- COBOL(COmmon Business Oriented Language)
  - Created in early 1960s
  - Primarily used for business, finance in companies and govenment
- BASIC(Beginner's All-purpose Symbolic Instruction Code)
  - Easy to learn and use for beginners

High Level Programming Language

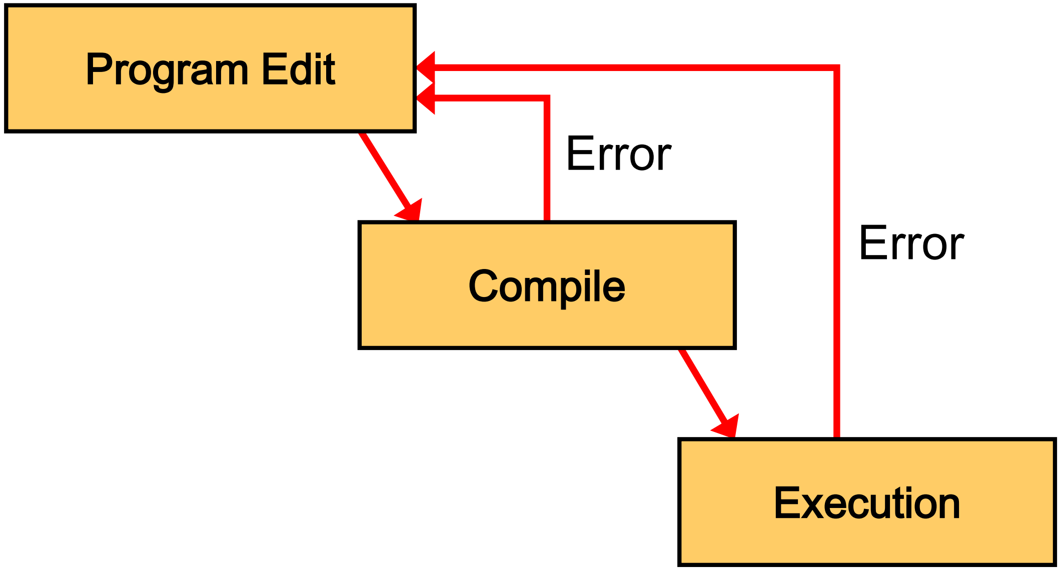
- C
  - Made by Dennis Ritchie (AT&T Bell Lab)
  - made for developing UNIX OS (1970s)
  - High level language with low level language properties (pointers,…)

- C++
  - Made by B. Stroustrup (AT&T Bell Lab)
  - OOPL(Object Oriented Programming Language) extending C

- Java
  - Made by James Gosling (Sun Microsystems, 1990s)
  - Platform independent OOPL

Programming and Execution

- **Programming Tool**
  - □ Editor, Compiler, Interpreter, Debugger, and etc
  - □ Integrated Development Environment **(IDE)**



Error

- **compile-time error**
  - Error occurring during compilation
  - Grammar check
  - Cannot execute if there is compile error

- **logical error**
  - Grammar is OK but logical error

- **run-time error**
  - Abnormal termination owing to unexpected reasons during program execution
  - Ex) divided by zero, illegal memory access

Debugging

- **debugging**
  - Bug : program error
  - Debugging : bug correction

Compiler / Interpreter

- Compiler
  - Convert high level language to low level language (occur at compile-time)
- Interpreter
  - Compile and execute the program line by line (occur at run-time)
- Comparison?