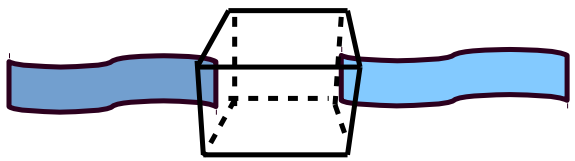
Fundamentals of Computer Programming

Lecture 1: An Introduction and Overview on C



Lecturer: Ahmad Siavashi

Spring 2023

Outline

Syllabus

1. Primitive Data Types and Operations
2. Sequential Control
3. Selection Control clause: if-else and switch
4. Loops Control clause: while, do-while and for
5. Functions: declaration, definition and calling
6. Pre-compilation Command/Macros: #ifdef
7. Array: declaration, definition and calling
8. Structures: struct and union
9. Pointers
10. File Operations: read and write

- Performance Evaluation
  - 20%×Midterm + 30%×Final
  - 25%×Exerc. + 20%×Proj. + 5%×Quiz.

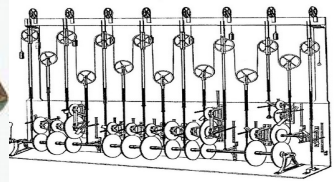
Arrangement of this course

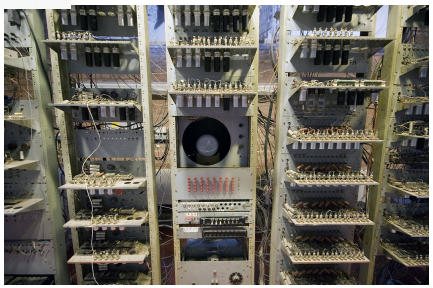
- 16 weeks $\times$ 1.5 hours classes
- Midterm exam on May 29, 2023
  - Live coding

Outline

About Computer (1)

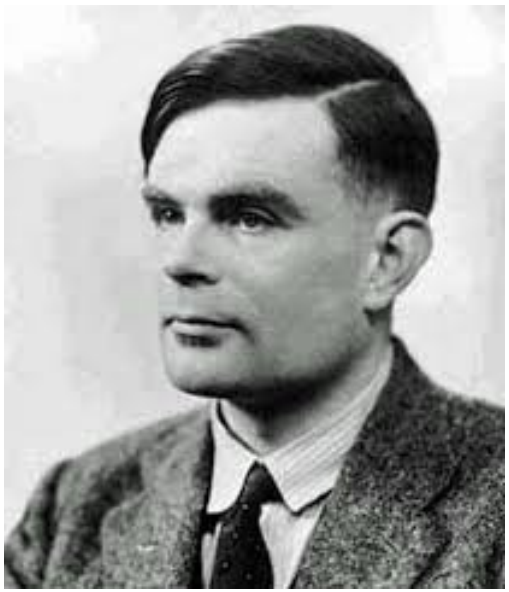
- What is computer?
  - Machine for computation
  - Essentially, no big difference from abacus
  - In our history, we have several kinds of machines used for computing
    - Abacus
    - Difference engine
    - Tide-predicting machine

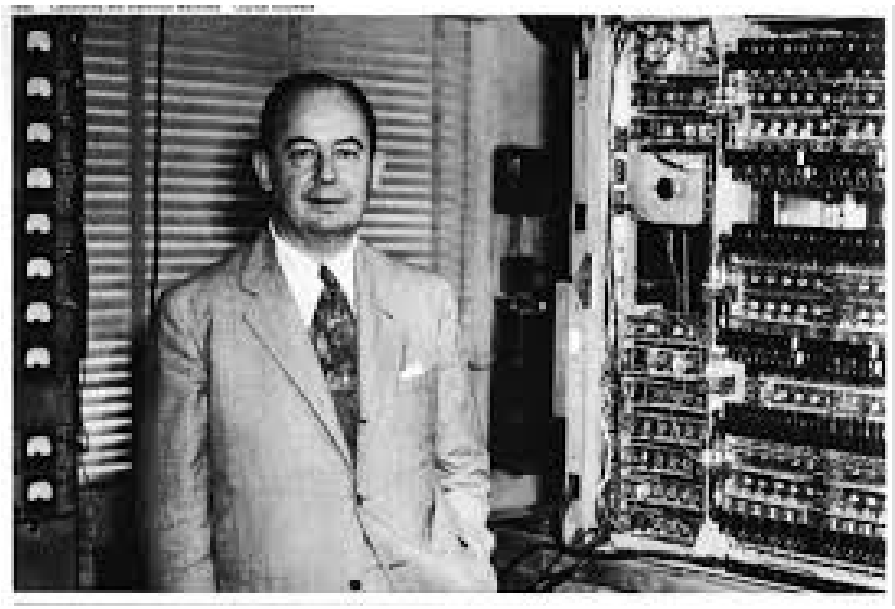


About Computer (2): the model

- What is computing
  - Input data and needed operations
  - Output the answer
  - This is actually the model proposed by **Alan Turing**



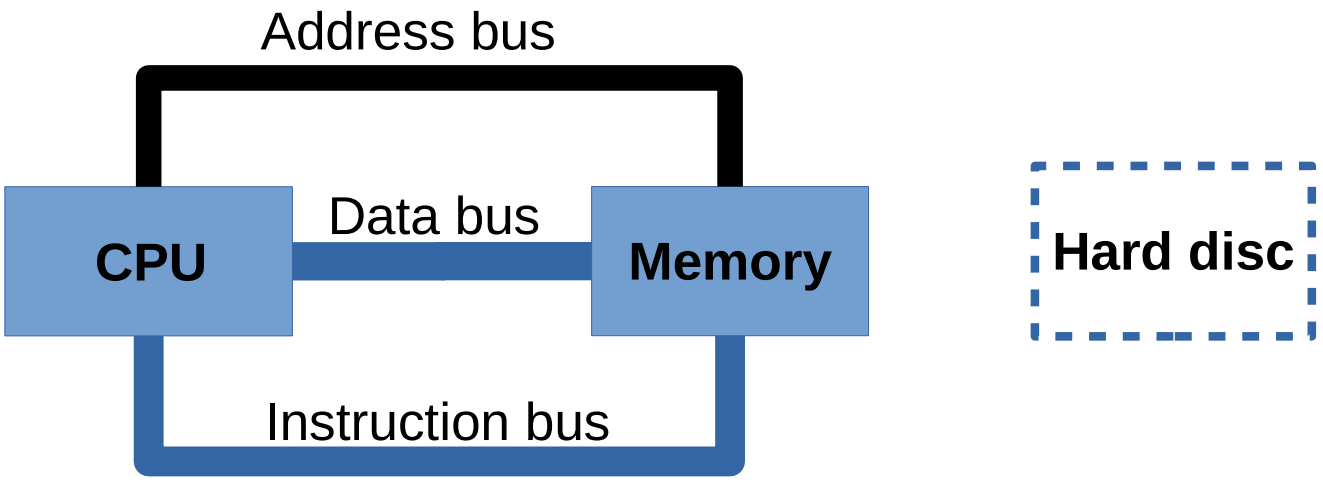
Alan Turing
(1912-1954)



John Von Neumann
(1903-1957)

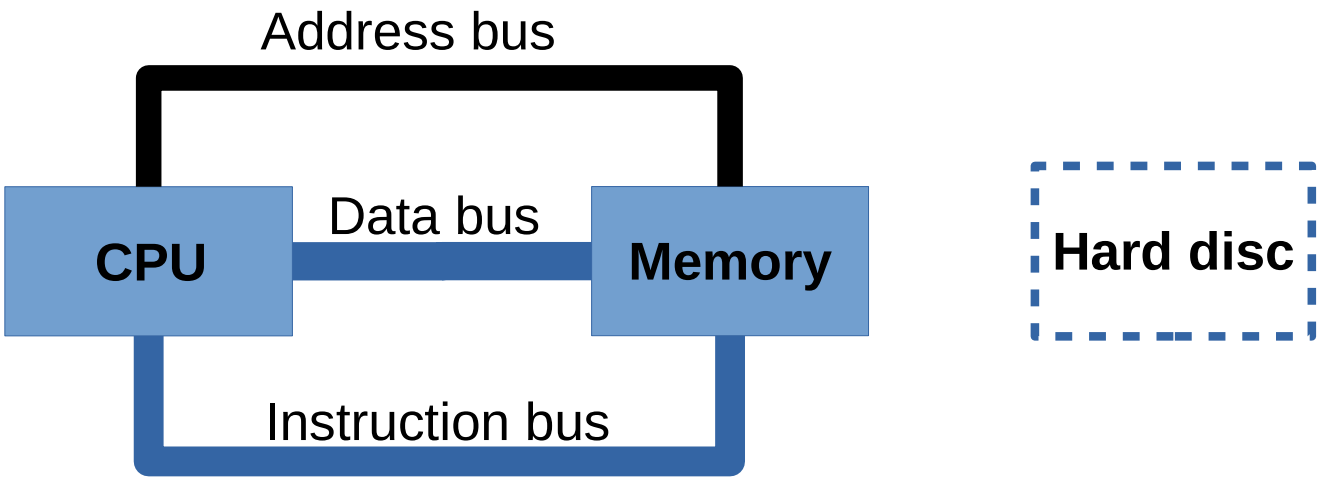
About Computer (3): the framework

- Think aloud about the major components of a computer
  - CPU: central processing unit
  - Memory
  - Hard disk
  - Keyboard
  - Graphics card + monitor/screen
  - Sound card + microphone + speaker
  - Mouse



About Computer (4): the framework

- Think aloud about the major components of a computer
  - **CPU: central processing unit**
  - **Memory**
  - Hard disk
  - Keyboard
  - Graphics card + monitor/screen
  - Sound card + microphone + speaker
  - Mouse



About Computer (5): who is who

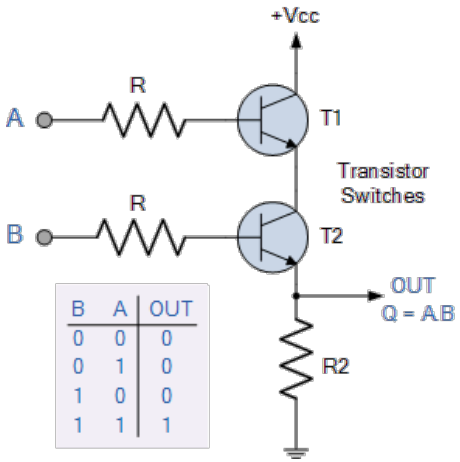




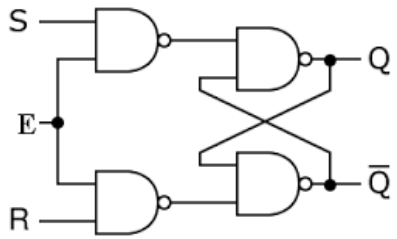


- How many of them you can figure out?

About Computer (6): basic elements in Computer Chips



AND Gate



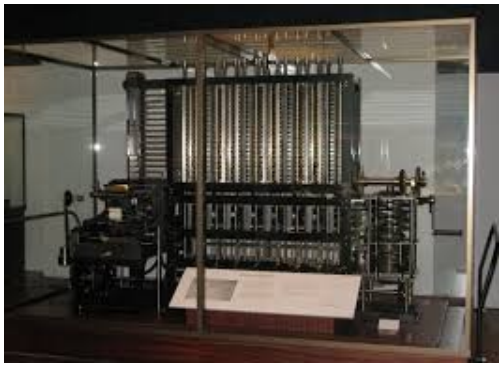
S-R Latch

- Despite the high complexity of VLSI (very large-scale integration)
- Two basic elements are there
- One is gate, responsible for operations, main components for CPU
- Another is latch, in charge of memory, main components for memory

Outline

Why programming? (1)

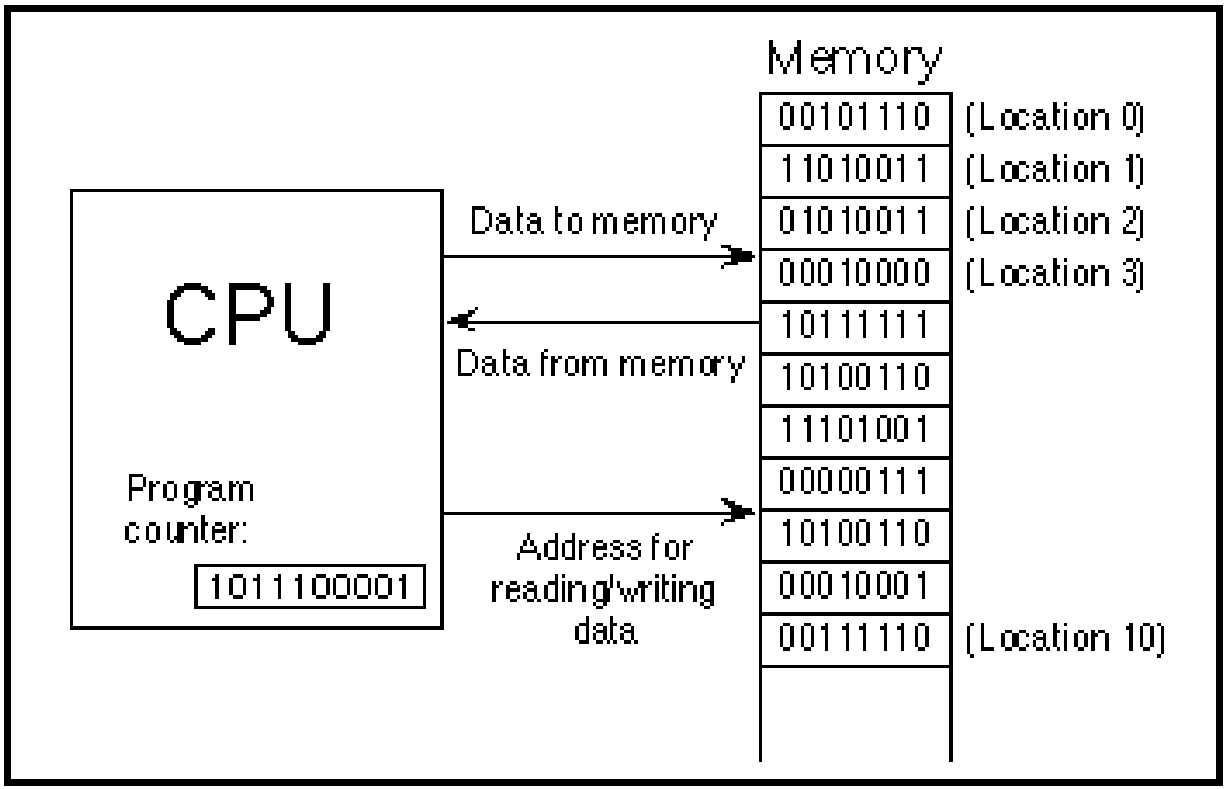
Charles Babbage
(1792—1871)



Mechanical computer

Ada Lovelace
(1815-1852)

Why programming? (2)

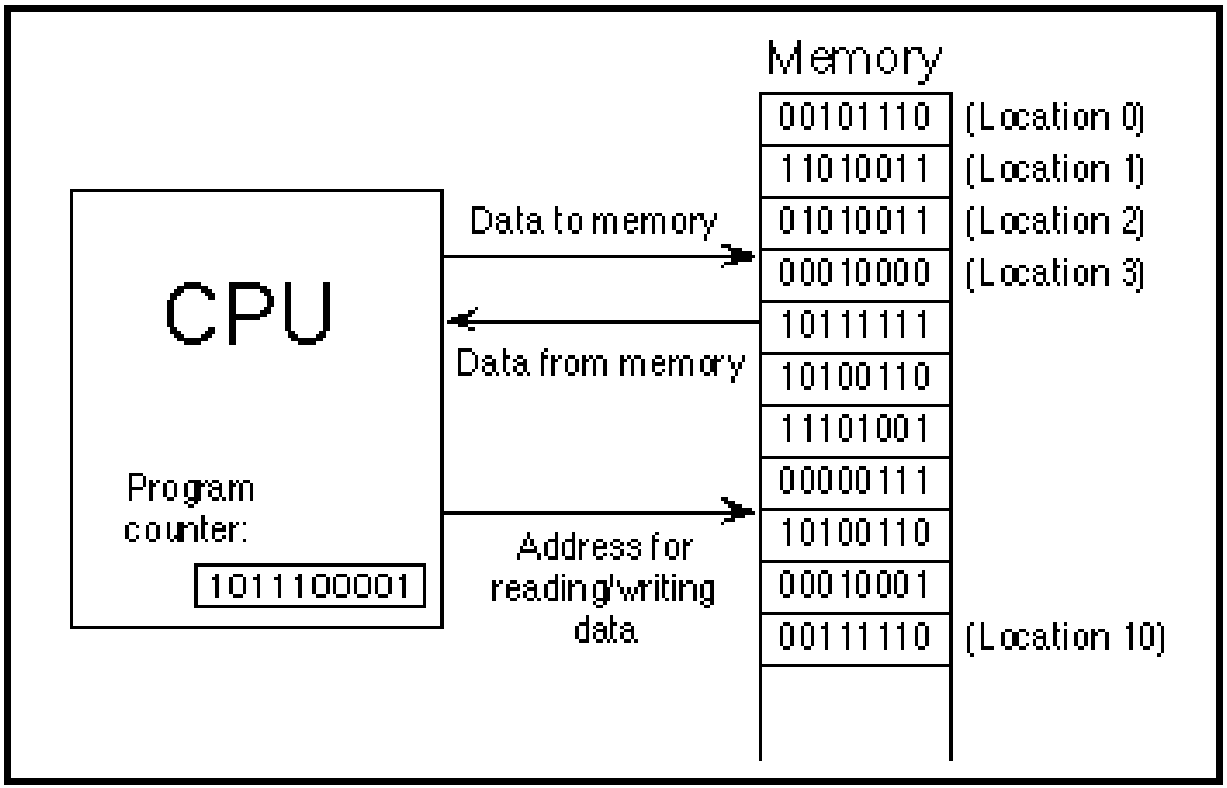


- Instructions and data fetch from memory to CPU for processing
- The results are returned back to memory

Why High Level Programming Language? (1)

- Natural language is the media that we communicate with each other
- Computer language is the media that we communicate with computer
- We should use the language that computer could understand
- At least, we need an interpreter/translator

Why High Level Programming Language? (2)



- Instructions are binary codes
- Machine only accepts/understands binary codes

Why Programming Language? (3)

1. 010101 0000 0011
2. 010101 0001 0101
3. 101010 0000 0001
4. 010101 0000 1011

Why Programming Language? (4)

1. 010101 0000 0011
2. 010101 0001 0101
3. 101010 0000 0001
4. 010101 0000 1011

1. MOV D1 0011
2. MOV D2 0101
3. ADD D1 D2
4. MOV D1 A1

- For the convenience of operation, binary instructions are denoted with readable symbols

Why Programming Language? (5)

- Machine code

1. 010101 0000 0011
2. 010101 0001 0101
3. 101010 0000 0001
4. 010101 0000 1011

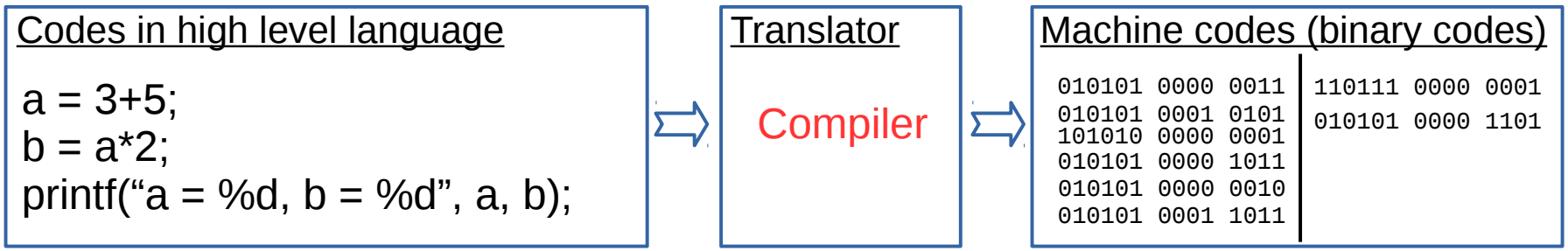
- Assembly

1. MOV D1 0011
2. MOV D2 0101
3. ADD D1 D2
4. MOV D1 A1

- High level language

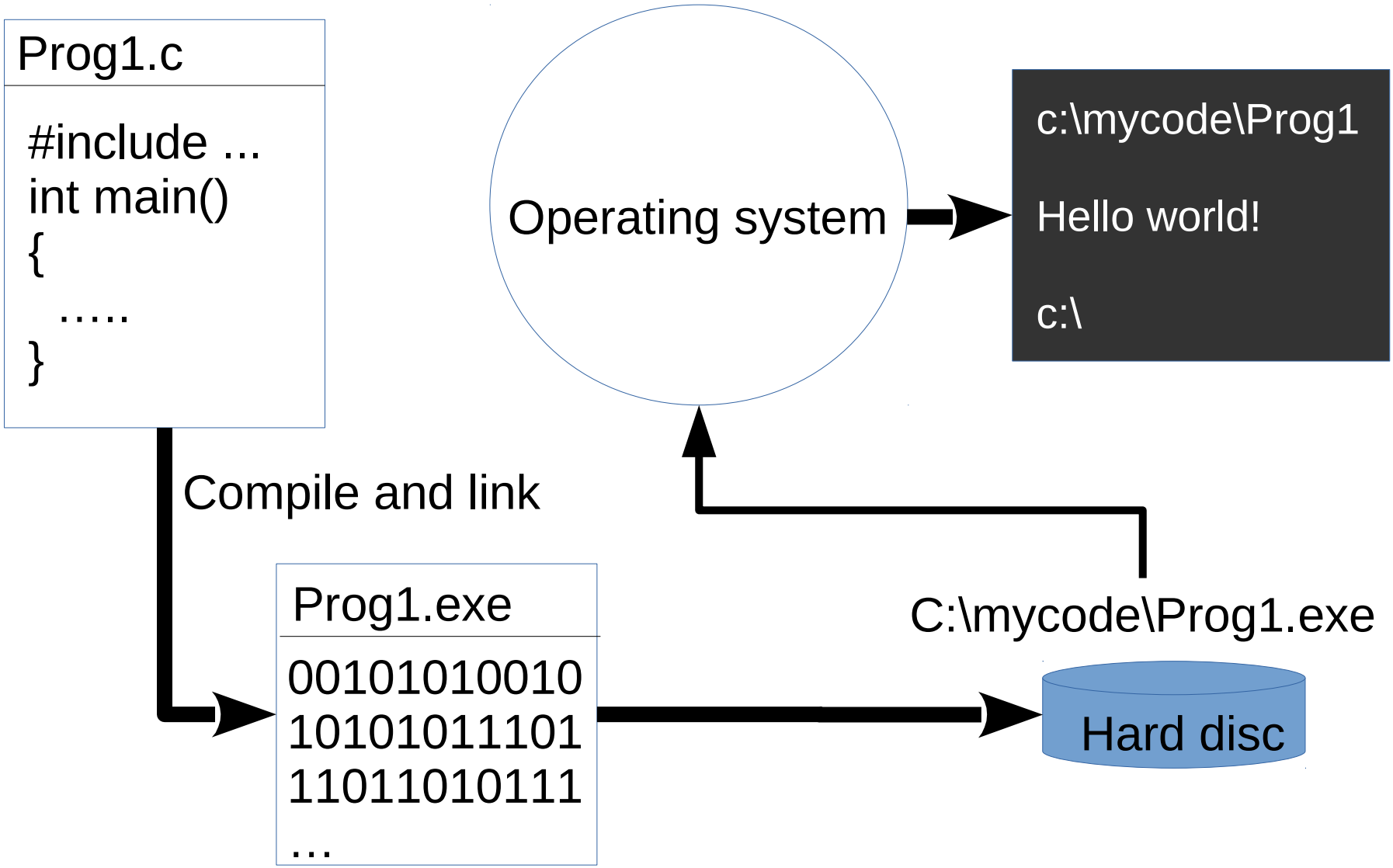
1. a=3+5;

Why Programming Language? (6)



- We write a text file in specified format (grammar)
- These are instructions that we basically understand
- The translator converts the text instructions into machine codes
- Machine then runs these binary codes one by one
- Different translators lead to different programming languages
- Which also regulate different grammars
- C is such kind of high level language

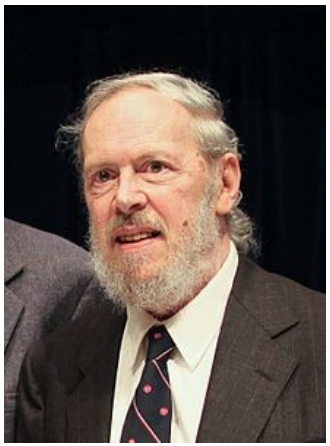
The life-time of a computer program



Outline

Brief History about C

Ken Thompson
(1943 -)



Dennis M. Ritchie
(1941 - 2011)

- C is born in AT&T Bell Labs along with UNIX
- The developer Dennis Ritchie and Ken Thompson were awarded with Turing Award
- C is simple:), versatile and highly efficient (70% of assembly language efficiency)
- UNIX is one of the most stable operating systems so far developed

Your first program in C (1)

```c
#include <stdio.h>
int main()
{  /*start of a block*/
    printf("Hello world!\n"); /*call function 'printf'*/
    return 0;                 /*return '0' back*/
} /*end of a block*/
```

- "#include <stdio.h>" states that we want to use function defined in "stdio.h"
- Our code is encapsulated in a function called "**main()**"
- In the main bordy of the function
- We output "Hello world!" to the screen
- "printf()" is a function declared in "stdio.h"
- include, int and return are reserved keywords

Your first program in C (2)

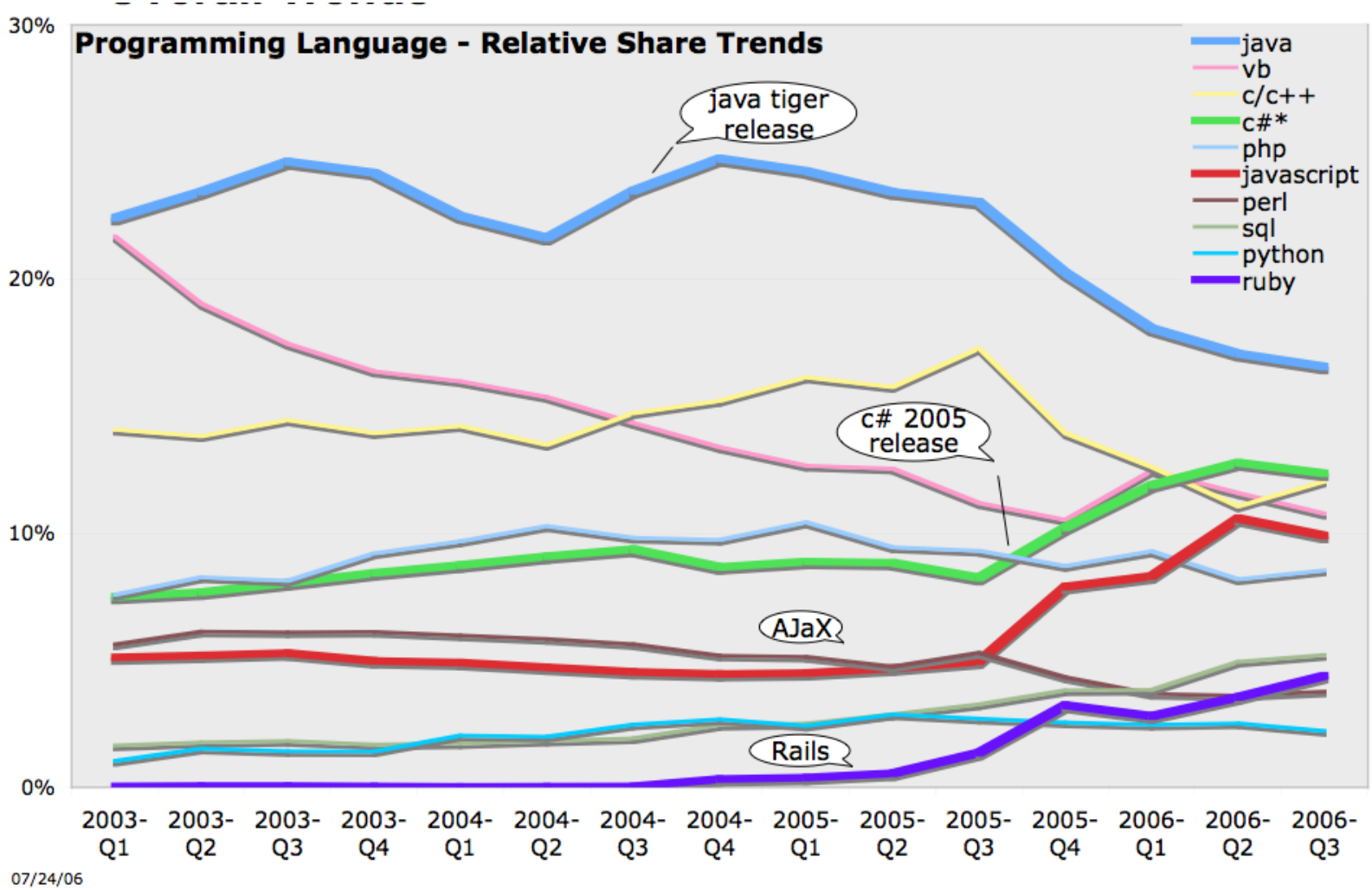
```c
#include <stdio.h>
int main()
{
    printf("Hello world 1!\n");
    printf("Hello world 2!\n");
    printf("Hello world 3!\n");
    return 0;
}
```

[Output]

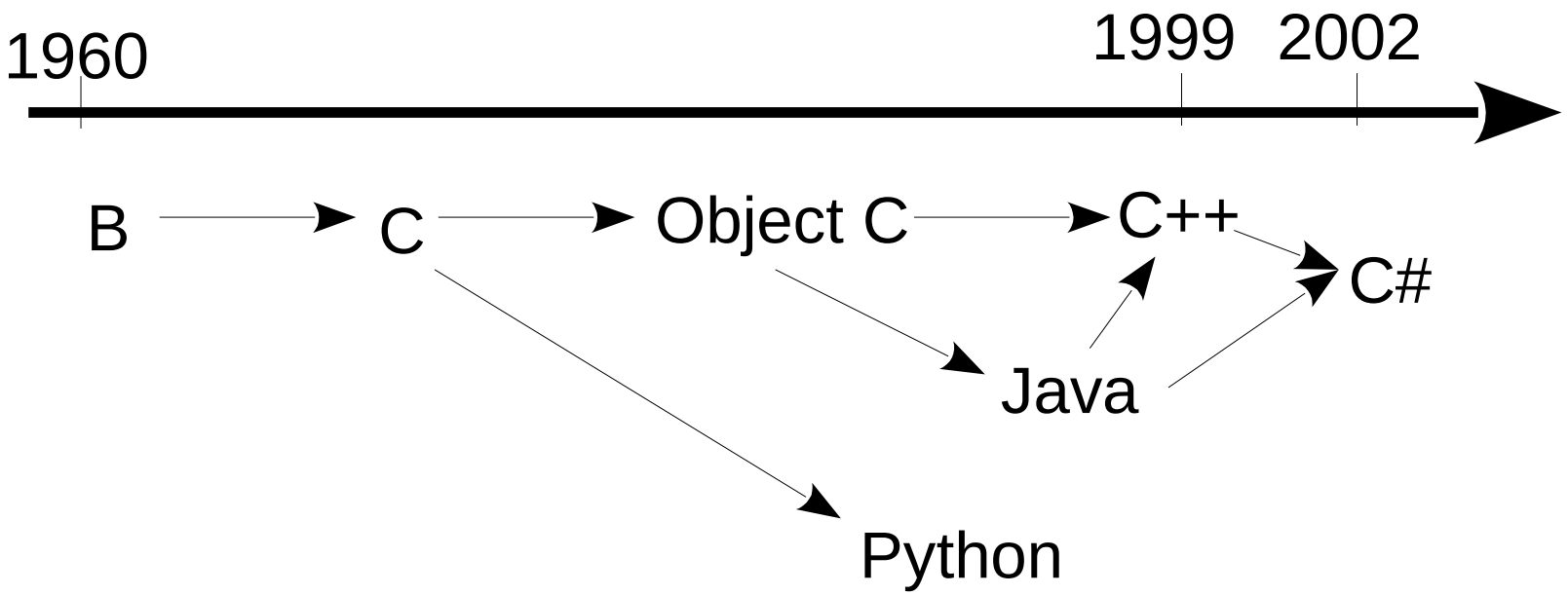
```
Hello world 1!
Hello world 2!
Hello world 3!
```

- Codes are executed from top to bottom

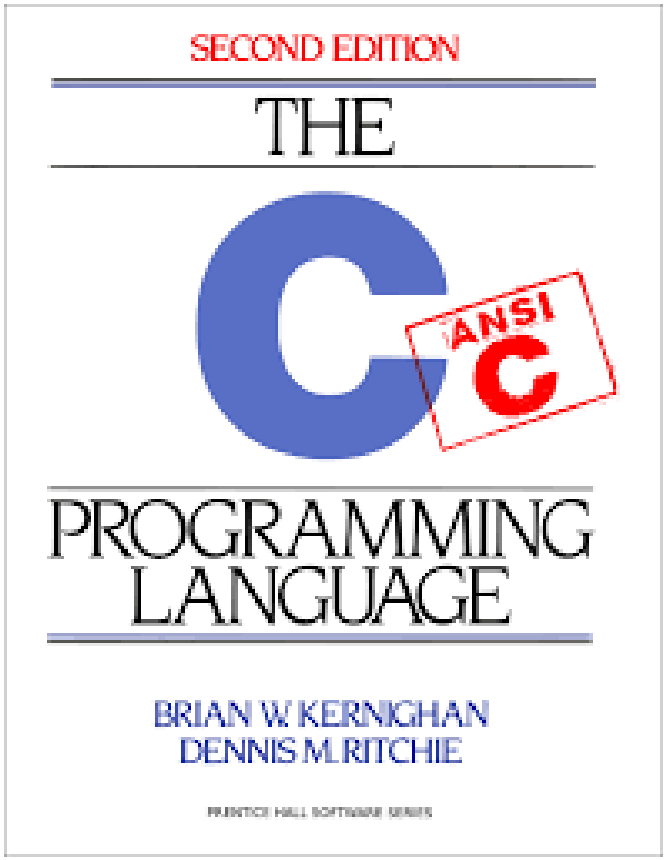
Popularity of C in recent decade



Popularity of C in recent decade



References



+

"C Programming" course slides by Dr. Wan-Lei Zhao from China