

DEMO FOR COMPUTER SYSTEM OVERVIEW

% YUZHE (RICHARD) TANG

A HELLO-WORLD PROGRAM

```
#include <stdio.h>

int main()
{
    printf("hello, world\n");
}
```

LIFETIME OF THE HELLO PROGRAM

1. source file
2. compilation
3. executable file
4. program execution
5. result

SOURCE FILE (1)

- text file
 - ASCII characters
 - program of C statements
 - tools: *vim* for text file editing
 - C/C++ language syntax

EXECUTABLE FILE (3)

- object file (binary file)
 - machine-language instructions
- file
 - storing a string of **bits**
 - **context** decides interpretation of bits

COMPILATION (2)

```
gcc hello.c -o a.out  
gcc -S hello.c -o hello.s #compiler  
gcc -c hello.s -o hello.o #assembler  
gcc hello.o -o a.out #linker
```

- compilation system
 - tools: *gcc/gdb* for compiling and debugging
 - 1. **preprocessor**: from source file to source
 - 2. **compiler**: from source to assembly file
 - *assembly file*
 - 3. **assembler**: from assembly file to relocatable object file
 - 4. **linker**: from multiple objects to an executable object
- why learn compiler internal?
 - security (buffer overflow), performance (pointer vs array index).

PROGRAM EXECUTION (4)

```
./a.out
```

- Program-execution overview
 1. the system loads executable a.out: from disk to main memory
 2. the system executes the a.out file in memory

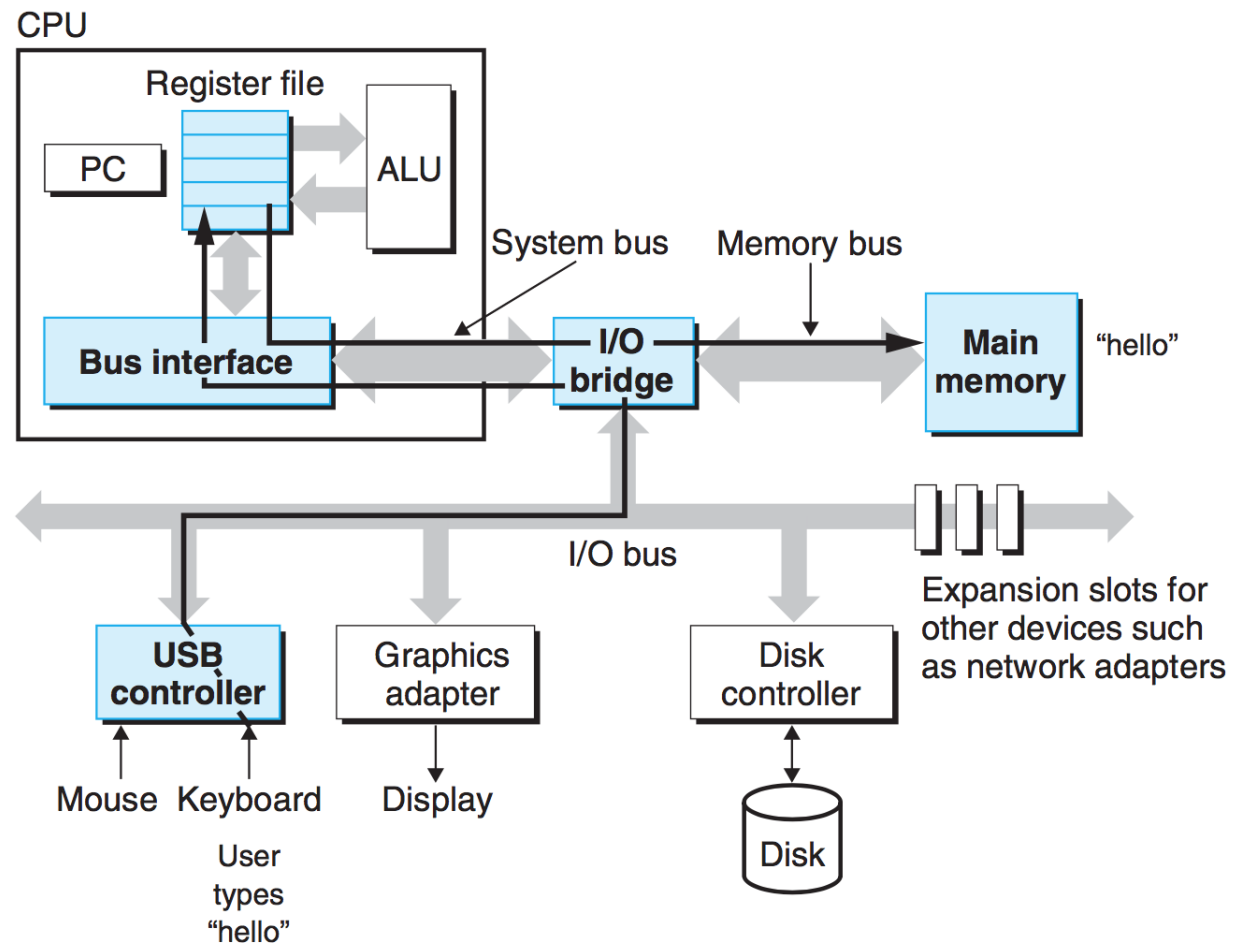
CPU EXECUTION MODEL

- model of cpu executing instructions
 - **CPU** reads an instruction from **virtual memory**,
 - executes it
 - and moves to the next instruction

COMPUTER SYSTEM OVERVIEW

Figure 1.5

Reading the hello command from the keyboard.



REFERENCES

- "Computer Systems: A Programmer's Perspective," Randal E. Bryant and David R. O'Hallaron, Chapter 1