

Introduction to Computer Systems

Lecture 1 – Introduction

2022 Spring, CSE3030

Sogang University



Team

- Instructor
 - Prof. Joo Ho Lee (AS818, jhleecs@sogang.ac.kr, 02-705-8489)
- Teaching assistants
 - Class I
 - Tae-young Eun, tyen7@gmail.com
 - Sejeong Oh, wwen9262@gmail.com
 - Class II
 - Yoochan Jeon, saycruder123@gmail.com
 - Junyong Kang, kang789082@gmail.com

Contents

Goal: Develop an general understanding of the underlying computer systems and improve system programming skills

- Data representation and arithmetic
- Machine-level program representation
- Processor architecture
- Linking
- Control flow
- Virtual memory
- System-level IO
- Network programming
- Concurrent programming

Ints are not Integers, Floats are not Reals

- **Example 1: Is $x^2 \geq 0$?**

- Float's: Yes!
- Int's:
 - $40000 * 40000 = 1600000000$
 - $50000 * 50000 = ??$

- **Example 2: Is $(x + y) + z = x + (y + z)$?**

- Unsigned & Signed Int's: Yes!
- Float's:
 - $(1e20 + -1e20) + 3.14 \rightarrow 3.14$
 - $1e20 + (-1e20 + 3.14) \rightarrow ??$

```
1  #include <stdio.h>
2
3  int main(){
4
5      // multiplication of int representation
6      int a = 40000;
7      int b = 50000;
8      printf("%d\n", a*a );
9      printf("%d\n", b*b );
10
11     // addition of float representation
12     printf("%f\n", (1e20+ -1e20) + 3.14 );
13     printf("%f\n", 1e20+ (-1e20 + 3.14) );
14     return 0;
15
16 }
```

```
vcg@vcg:~$ ./a.out
1600000000
-1794967296
3.140000
0.000000
vcg@vcg:~$
```

Memory Referencing Bug Example

```
typedef struct {  
    int a[2];  
    double d;  
} struct_t;  
  
double fun(int i) {  
    volatile struct_t s;  
    s.d = 3.14;  
    s.a[i] = 1073741824; /* Possibly out of bounds */  
    return s.d;  
}
```

```
fun(0)    ->    3.14  
fun(1)    ->    3.14  
fun(2)    ->    3.1399998664856  
fun(3)    ->    2.00000061035156  
fun(4)    ->    3.14  
fun(6)    ->    Segmentation fault
```

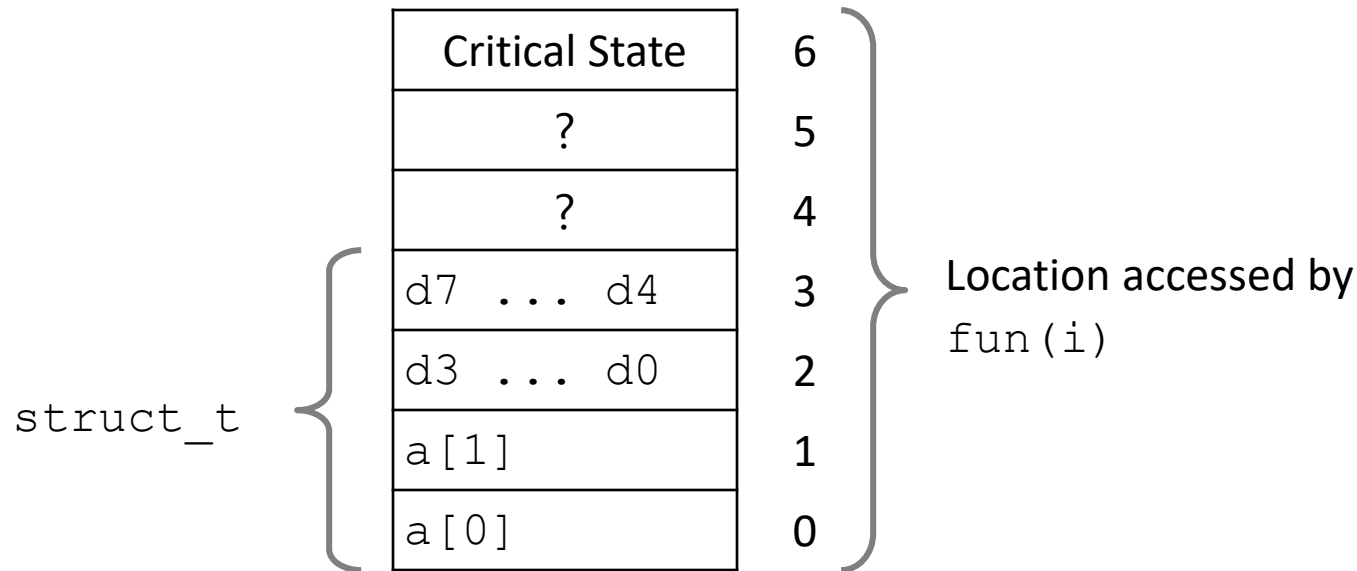
- Result is system specific

Memory Referencing Bug Example

```
typedef struct {  
    int a[2];  
    double d;  
} struct_t;
```

fun(0)	->	3.14
fun(1)	->	3.14
fun(2)	->	3.1399998664856
fun(3)	->	2.00000061035156
fun(4)	->	3.14
fun(6)	->	Segmentation fault

Explanation:



Copy rows

```
void copyij(int src[2048][2048],
            int dst[2048][2048])
{
    int i,j;
    for (i = 0; i < 2048; i++)
        for (j = 0; j < 2048; j++)
            dst[i][j] = src[i][j];
}
```

4.3ms

Copy columns

```
void copyji(int src[2048][2048],
            int dst[2048][2048])
{
    int i,j;
    for (j = 0; j < 2048; j++)
        for (i = 0; i < 2048; i++)
            dst[i][j] = src[i][j];
}
```

81.8ms

2.0 GHz Intel Core i7 Haswell

- Hierarchical memory organization
- Performance depends on access patterns
 - Including how step through multi-dimensional array

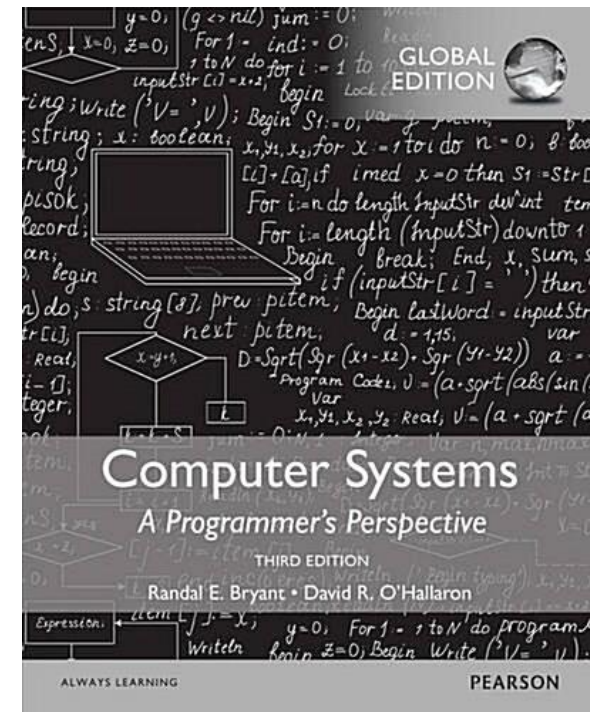
Course Perspective

- Our Course is Programmer-Centric
 - Purpose is to show that by knowing more about the underlying system, one can be more effective as a programmer
 - Enable you to
 - Write programs that are more reliable and efficient
 - Incorporate features that require hooks into OS
 - E.g., concurrency, signal handlers
 - Cover material in this course that you won't see elsewhere
 - Not just a course for dedicated hackers
 - **We bring out the hidden hacker in everyone!**

Textbooks

“Computer systems: A programmer’s perspective”, 3rd edition

- Randal E. Bryant and David R. O’Hallaron
- <http://csapp.cs.cmu.edu/3e/students.html>
- Lecture materials and class formats are referred to and improved from
 - Official lecture slides



Organization

- 3 Credits
- Lectures held via **Zoom**
 - Class I: Tuesday and Thursday 9:00-10:15
 - Class II: Wednesday and Friday 9:00-10:15
- Exam
 - Date and time will be announced on the cyber campus.
 - Relevant to programming assignments and pen & paper assignments.

Exercises

- We offer total 7 assignments
 - **3 programming assignments + 4 pen & paper assignments**
- Handed out via the cybercampus.
- **Up to 2 students** can perform the assignments.
- Assignments are **not graded**, but we will give support and solutions.
- They are highly **relevant to the exam**.

Programming Assignments

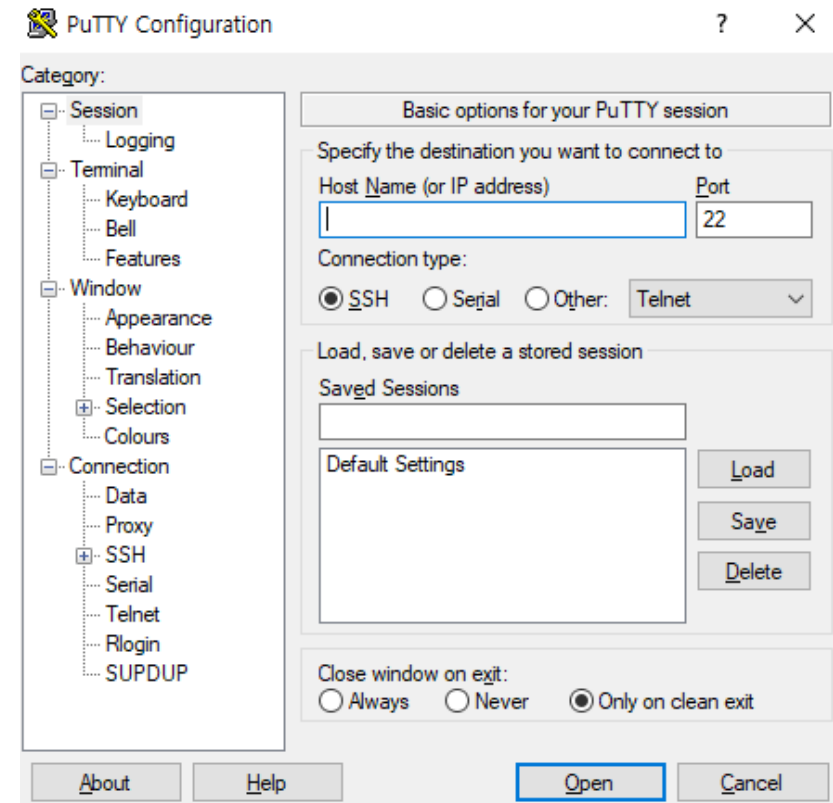
- Exercises
 - PA 1: Compiler and assembler
 - PA 2: Shell program
 - PA 3: Shell program via network
- You will have 3-4 weeks per PA and submit via CyberCampus.
- We will provide additional videos about basic instructions for programming.
- Submit your code only **(NOT GRADED!)**

How to make a team?

- Step 1: find a team member
 - Contact your friend or find someone via the cyber campus.
 - Or, you can do it alone.
- Step 2: send the team information to a TA
 - Team name
 - Member's name and id
 - TA emails
 - Yoochan Jeon, saycruder123@gmail.com
 - Junyong Kang, kang789082@gmail.com

Facilities

- For each team will get an team id of the ubuntu server.
 - Address: <http://cspro.sogang.ac.kr/>
- How to connect
 - SSH connection
 - ssh [id@cspro.sogang.ac.kr](http://cspro.sogang.ac.kr/)
 - Using putty
 - <https://www.putty.org/>



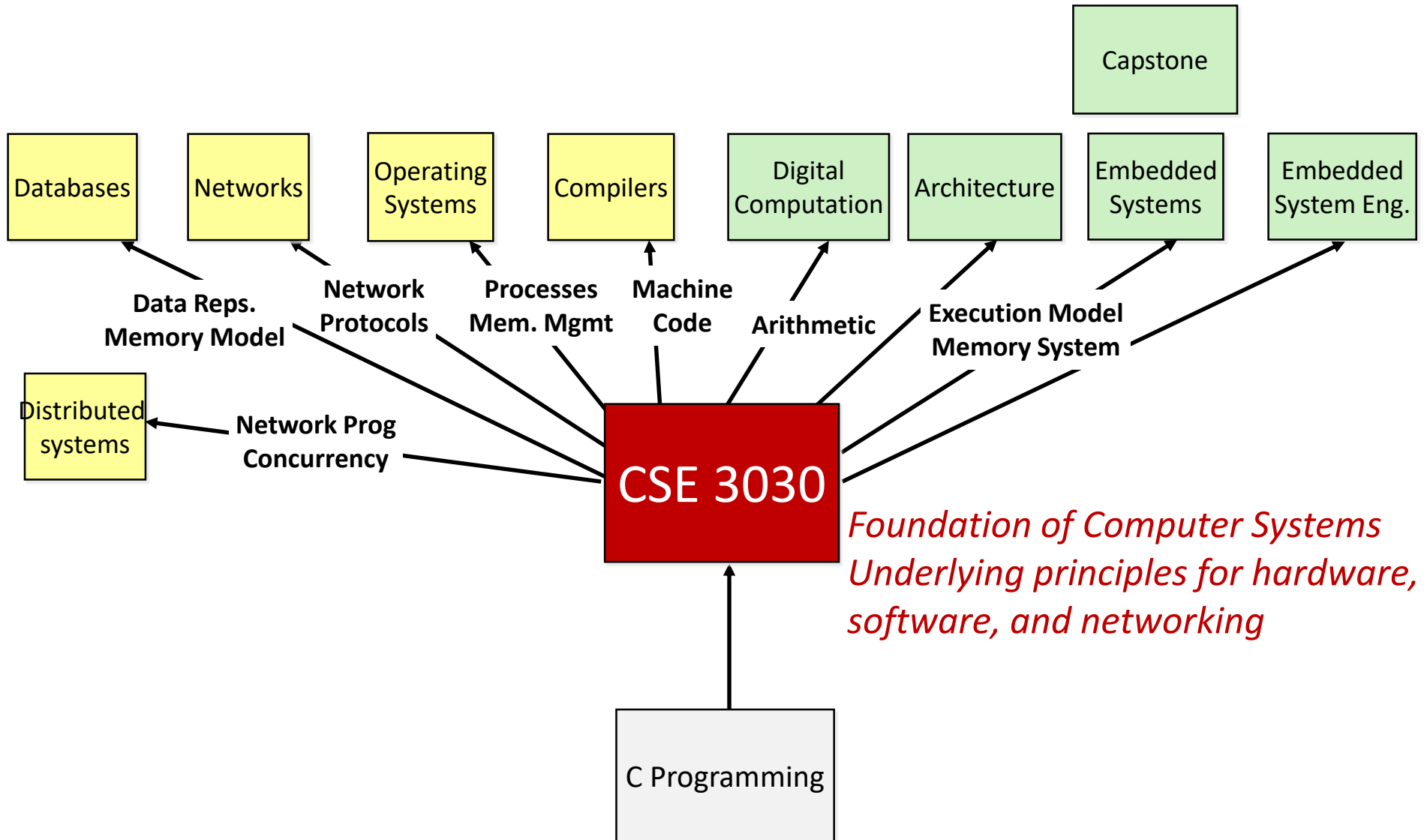
Grading

- Exam (95%)
 - Midterm & Final Exams (45% and 50%)
- Term Projects (0%)
 - 3 projects
 - **Relevant to exams**
- Homework (0%)
 - 4 pen & papers
 - Again **relevant to exams**
- Attendance (5%)
 - Your attitude and time record on the CyberCampus.

Cheating

- Penalty for cheating:
 - Removal from course with failing grade (no exceptions!)
 - Permanent mark on your record
 - Your instructors' personal contempt
- Exam modality
 - Online or offline (to be determined)
 - If it is online, we will give you how to set up your environment for taking the exam.

Role within CSE/EE Curriculum



Prerequisites

- **C programming**

- Data, variables, types, functions, loop statements, if statements etc.

Good luck