



Data Structures (Spring 2020)

Iteration & Recursion (2nd Lab)

2020.03.27

Seoul National University

Database Systems Lab

Today's Lab

- VirtualBox tips
- Eclipse tips
- Fibonacci numbers
- Tower of Hanoi



VirtualBox Tips

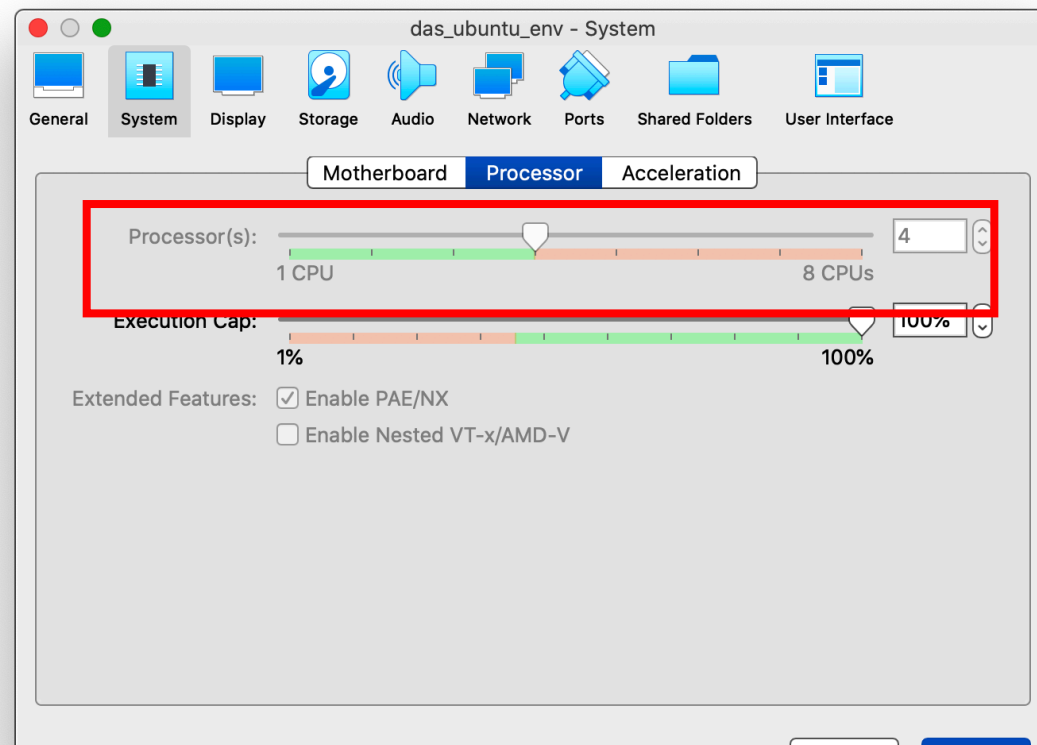
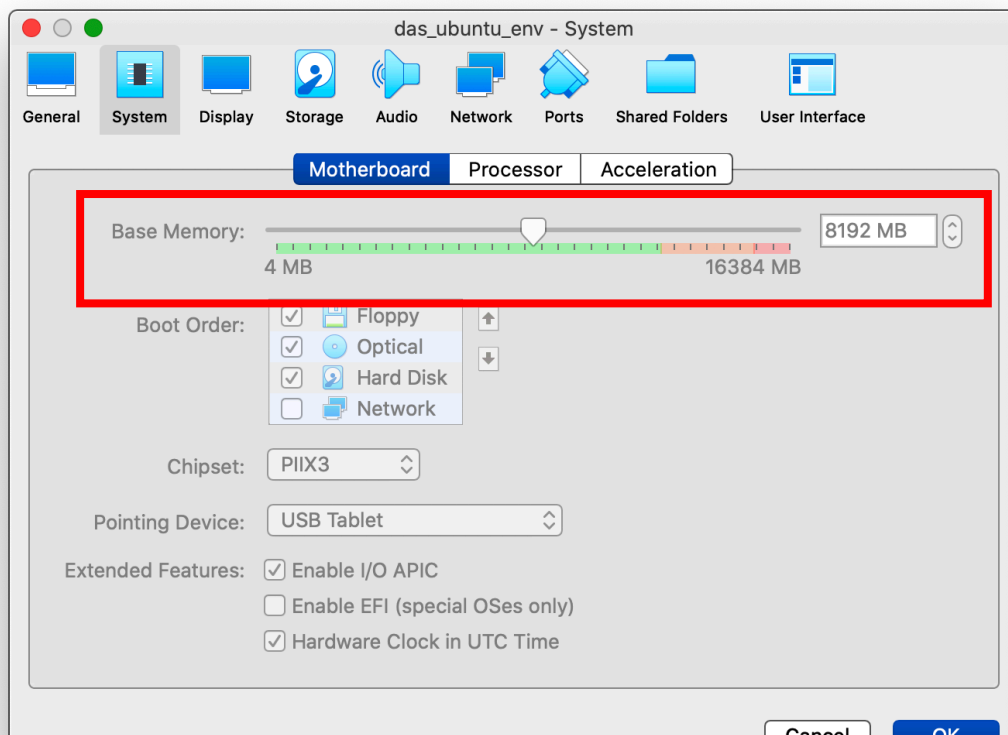
- Troubleshooting
 - My virtual machine is too slow
 - My display is too small

=> **Change your VirtualBox settings**



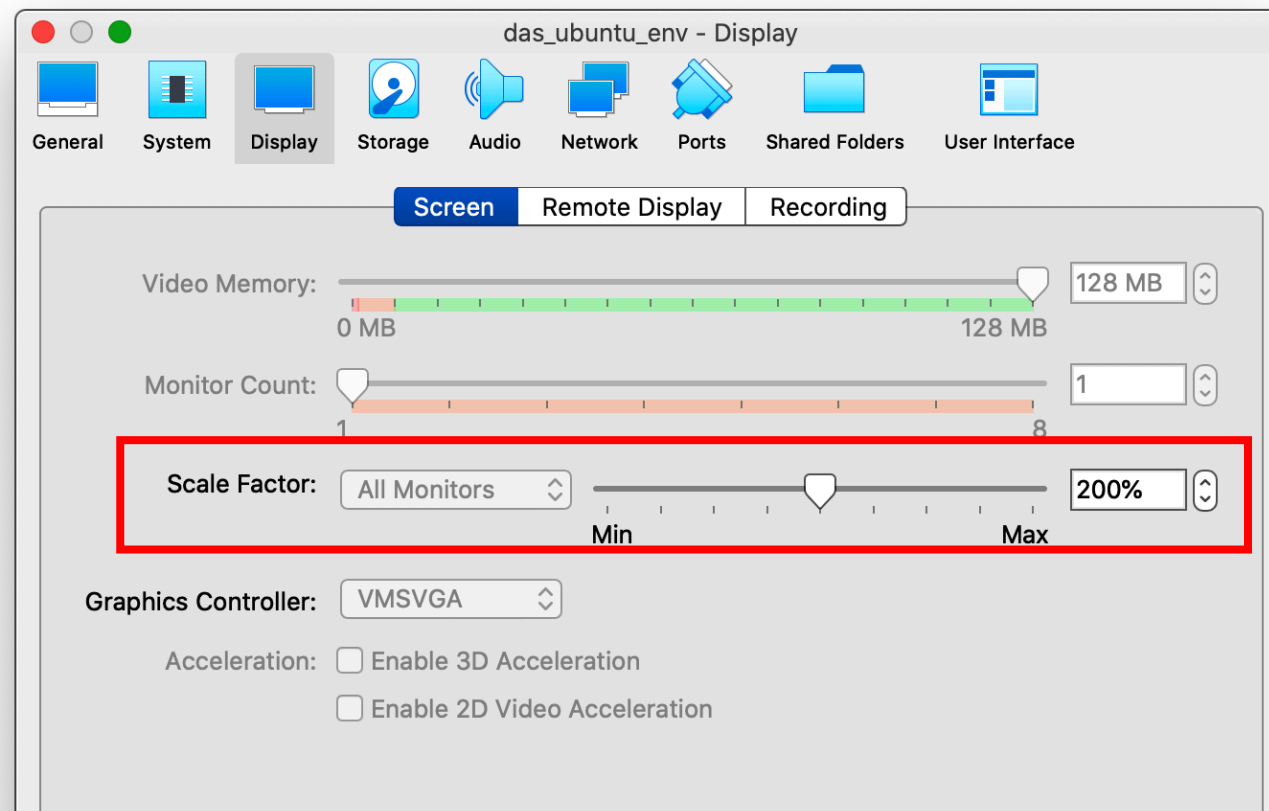
VirtualBox Tips

- My virtual machine is too slow
 - Setting -> System -> Motherboard -> Increase your "Processor"
 - Setting -> System -> Motherboard -> Increase your "Base Memory"



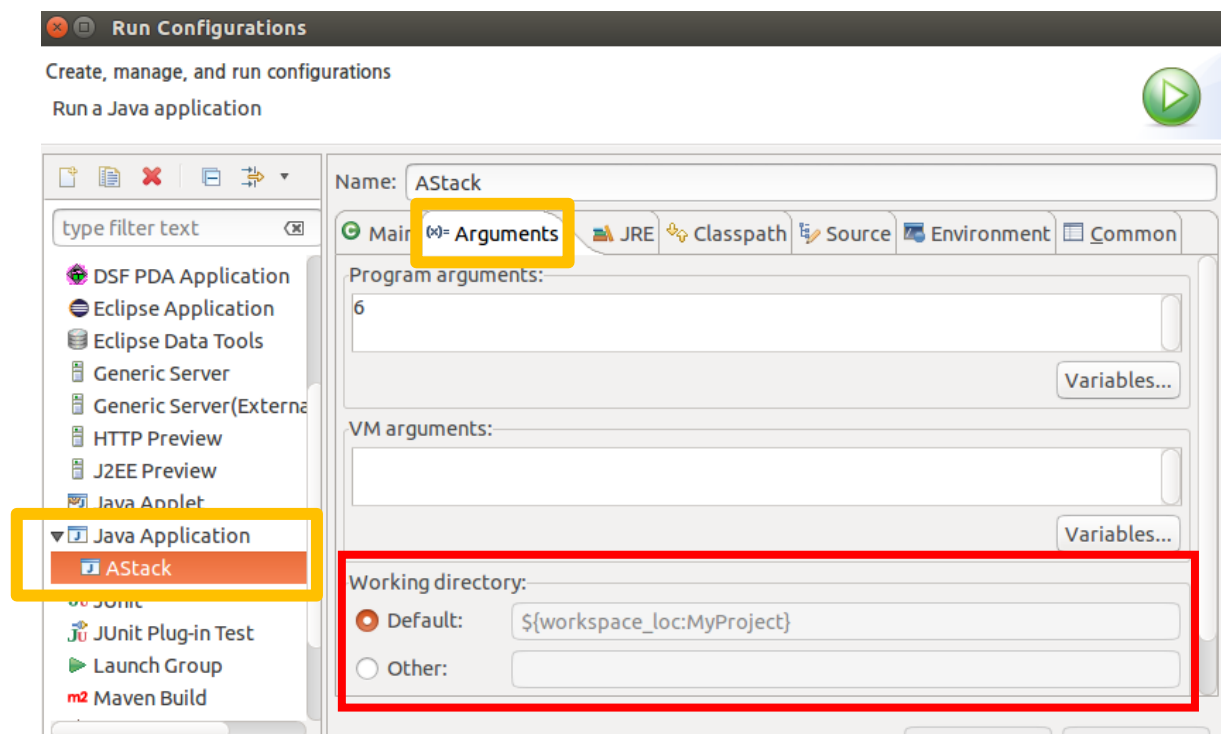
VirtualBox Tips

- My display is too small
 - Setting -> Display -> Screen -> Scale Factor : 200



Eclipse Tips

- If you want to write your program on Eclipse
 - Run As -> Run Configurations -> Arguments
 - Working directory -> Select "Other" -> choose your "public" directory



Eclipse Tips

- You can run your code on any OS!
- **BUT, you should check your code on Linux environment** when you submit your **program assignment**

Fibonacci numbers

- In mathematics, the **Fibonacci numbers**, commonly denoted F_n , form a sequence, called the Fibonacci sequence, such that each number is the sum of the two preceding ones, starting from 0 and 1. That is,

$$F_0 = 0, \quad F_1 = 1, \quad \text{and} \quad F_n = F_{n-1} + F_{n-2}, \quad \text{for } n > 1$$

- The beginning of the sequence is thus:

0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, ...

Tower of Hanoi

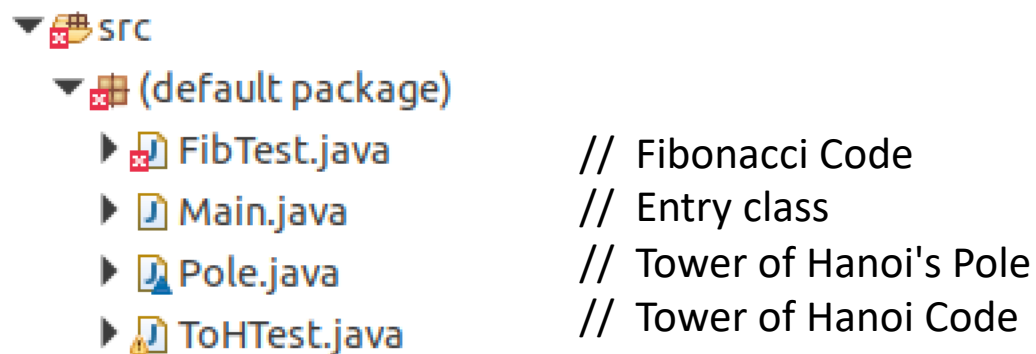
- The **Tower of Hanoi** is a mathematical game or puzzle.
 - three rods (or poles), a number of disks of different sizes
 - the puzzle starts with the disks in a neat stack in ascending order of size on one rod, the smallest at the top, thus making a conical shape.
- The objective of the puzzle is **to move the entire stack to another rod**, obeying **the following simple rules**:
 1. Only one disk can be moved at a time.
 2. Each move consists of taking the upper disk from one of the stacks and placing it on top of another stack or on an empty rod.
 3. No larger disk may be placed on top of a smaller disk.



From: https://en.wikipedia.org/wiki/Tower_of_Hanoi

Exercises

- Fill the blank of codes
 - Write your code into "// TODO:" section (FibTest.java, ToHTest.java)
 - to complete Fibonacci number and Tower of Hanoi



Project Structure

```

4
5 public class Main {
6     public static void main(String[] input) {
7         int num = Integer.parseInt(input[1]);
8         if (input[0].contentEquals("fib")) {
9
10            System.out.println("fibr("+num+") is " +FibTest.fibi(num));
11        } else if (input[0].contentEquals("toh")) {
12
13            ToHTest.TOH(num);
14        }
15    }
16 }
17
    
```

Main.java

Exercises

- Fill the blank of codes
 - Write your code into "// TODO:" section (FibTest.java, ToHTest.java)
 - to complete Fibonacci number and Tower of Hanoi

```
/**
 * Compute the moves to solve a Tower of Hanoi puzzle. Function move does (or
 * prints) the actual move of a disk from one pole to another.
 *
 * @param n      The number of disks
 * @param start  The start pole
 * @param goal   The goal pole
 * @param temp   The other pole
 */
void TOH(int n, Pole start, Pole goal, Pole temp) {
    // TODO:
}
```

ToHTest.java

```
static long fibi(int n) {
    // fibr(91) is the largest value that fits in a long
    // TODO:
}
```

FibTest.java



Exercises

- Result

\$ java Main fib 1
fibr(1) is 1

\$ java Main fib 2
fibr(2) is 1

\$ java Main fib 5
fibr(5) is 5

\$ java Main fib 10
fibr(10) is 55

\$ java Main fib 20
fibr(20) is 6765

\$ java Main fib 40
fibr(40) is 102334155

\$ java Main toh 2
1: Move 1 to 3
2: Move 1 to 2
3: Move 3 to 2

\$ java Main toh 3
1: Move 1 to 2
2: Move 1 to 3
3: Move 2 to 3
4: Move 1 to 2
5: Move 3 to 1
6: Move 3 to 2
7: Move 1 to 2