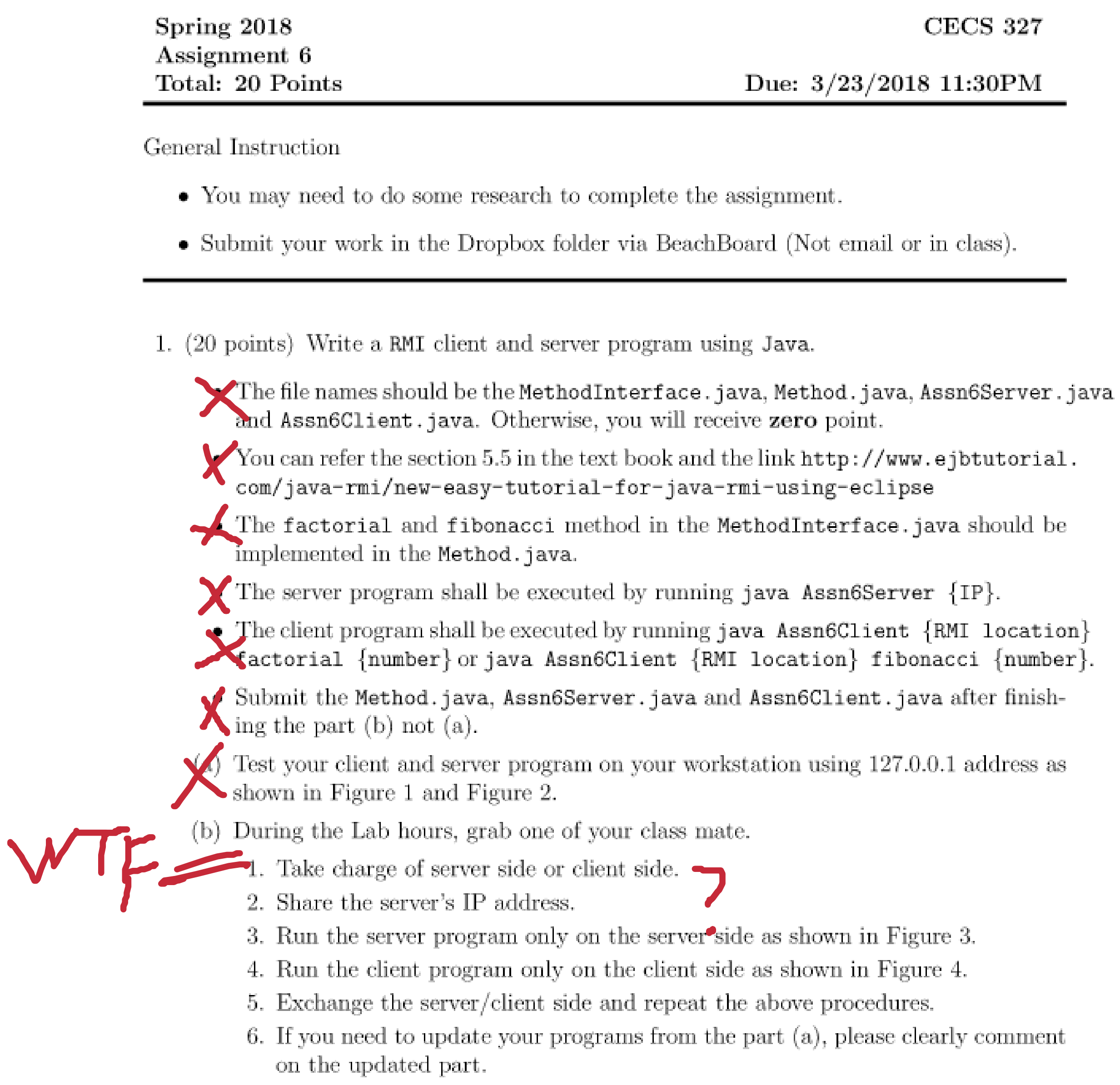
Lab 6 - RMI

Wednesday, March 14, 2018 11:01 AM



Spring 2018 **CECS 327**
Assignment 6
Total: 20 Points **Due: 3/23/2018 11:30PM**

General Instruction

- You may need to do some research to complete the assignment.
- Submit your work in the Dropbox folder via BeachBoard (Not email or in class).

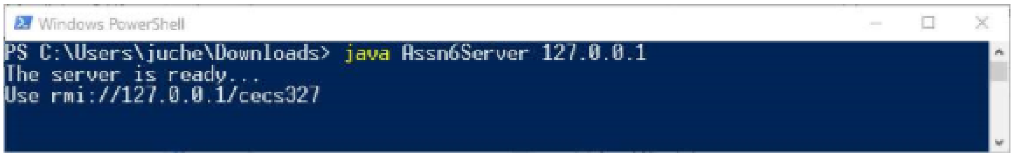
1. (20 points) Write a `RMI` client and server program using `Java`.
   - The file names should be the `MethodInterface.java`, `Method.java`, `Assn6Server.java` and `Assn6Client.java`. Otherwise, you will receive **zero** point.
   - You can refer the section 5.5 in the text book and the link `http://www.ejbtutorial.com/java-rmi/new-easy-tutorial-for-java-rmi-using-eclipse`
   - The `factorial` and `fibonacci` method in the `MethodInterface.java` should be implemented in the `Method.java`.
   - The server program shall be executed by running `java Assn6Server {IP}`.
   - The client program shall be executed by running `java Assn6Client {RMI location} factorial {number}` or `java Assn6Client {RMI location} fibonacci {number}`.
   - Submit the `Method.java`, `Assn6Server.java` and `Assn6Client.java` after finishing the part (b) not (a).

   (a) Test your client and server program on your workstation using 127.0.0.1 address as shown in Figure 1 and Figure 2.

   (b) During the Lab hours, grab one of your class mate.

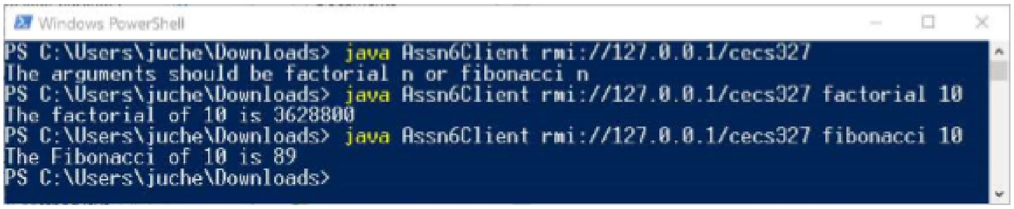
   WTF

   1. Take charge of server side or client side. ?
   2. Share the server's IP address.
   3. Run the server program only on the server side as shown in Figure 3.
   4. Run the client program only on the client side as shown in Figure 4.
   5. Exchange the server/client side and repeat the above procedures.
   6. If you need to update your programs from the part (a), please clearly comment on the updated part.



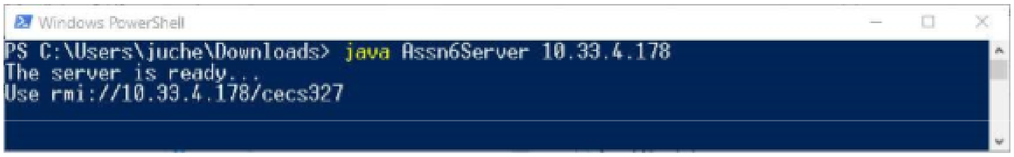
```
PS C:\Users\juche\Downloads> java Assn6Server 127.0.0.1
The server is ready...
Use rmi://127.0.0.1/cecs327
```

Figure 1: RMI server by using a single workstation



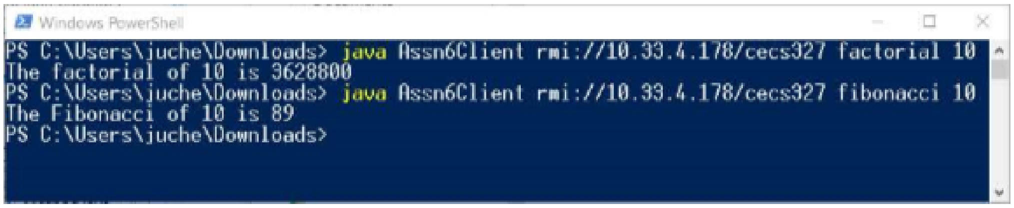
```
PS C:\Users\juche\Downloads> java Assn6Client rmi://127.0.0.1/cecs327
The arguments should be factorial n or fibonacci n
PS C:\Users\juche\Downloads> java Assn6Client rmi://127.0.0.1/cecs327 factorial 10
The factorial of 10 is 3628800
PS C:\Users\juche\Downloads> java Assn6Client rmi://127.0.0.1/cecs327 fibonacci 10
The Fibonacci of 10 is 89
PS C:\Users\juche\Downloads>
```

Figure 2: RMI client by using a single workstation



```
PS C:\Users\juche\Downloads> java Assn6Server 10.33.4.178
The server is ready...
Use rmi://10.33.4.178/cecs327
```

Figure 3: RMI server by using two workstations



```
PS C:\Users\juche\Downloads> java Assn6Client rmi://10.33.4.178/cecs327 factorial 10
The factorial of 10 is 3628800
PS C:\Users\juche\Downloads> java Assn6Client rmi://10.33.4.178/cecs327 fibonacci 10
The Fibonacci of 10 is 89
PS C:\Users\juche\Downloads>
```

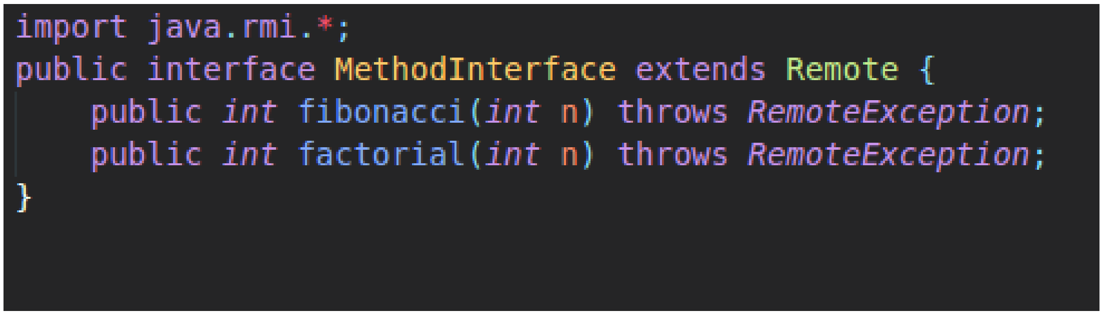
Figure 4: RMI client by using two workstations

SERVER SIDE:

For server side you need the following 3 files: **Assn6Server.java, MethodsInterface.java, and Methods.java**

For **MethodsInterface.java**

- You need to define the interface and the features Moon wants to you to implement such as the Fibonacci and Factorial methods
- Note name this file, MethodsInterface.java

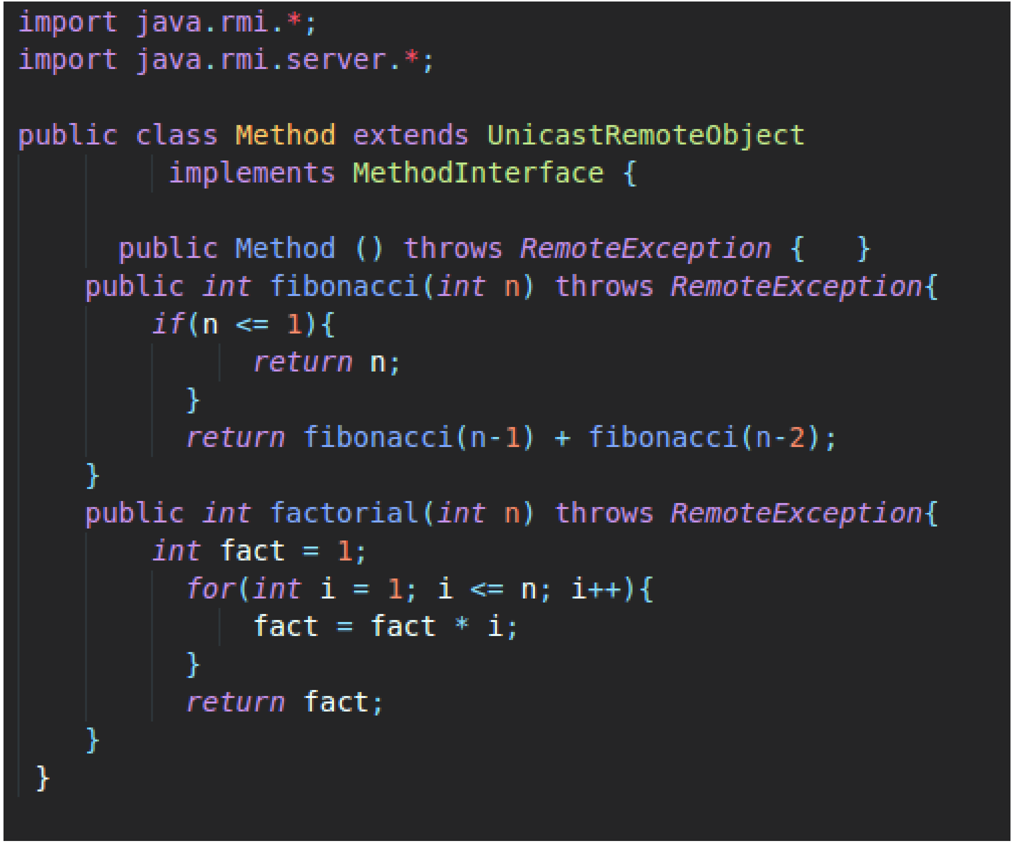


```
import java.rmi.*;
public interface MethodInterface extends Remote {
    public int fibonacci(int n) throws RemoteException;
    public int factorial(int n) throws RemoteException;
}
```

- Done

For **Methods.java**

- You need to actually implement the methods defined in the MethodsInterface.java file
- Google "Fibonacci Java Code" and "Factorial Java Code"
- This is one example I got:

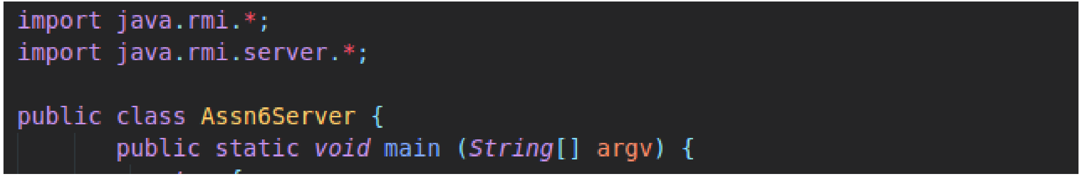


```
import java.rmi.*;
import java.rmi.server.*;

public class Method extends UnicastRemoteObject
        implements MethodInterface {

      public Method () throws RemoteException {   }
    public int fibonacci(int n) throws RemoteException{
        if(n <= 1){
            return n;
        }
        return fibonacci(n-1) + fibonacci(n-2);
    }
    public int factorial(int n) throws RemoteException{
        int fact = 1;
        for(int i = 1; i <= n; i++){
            fact = fact * i;
        }
        return fact;
    }
 }
```

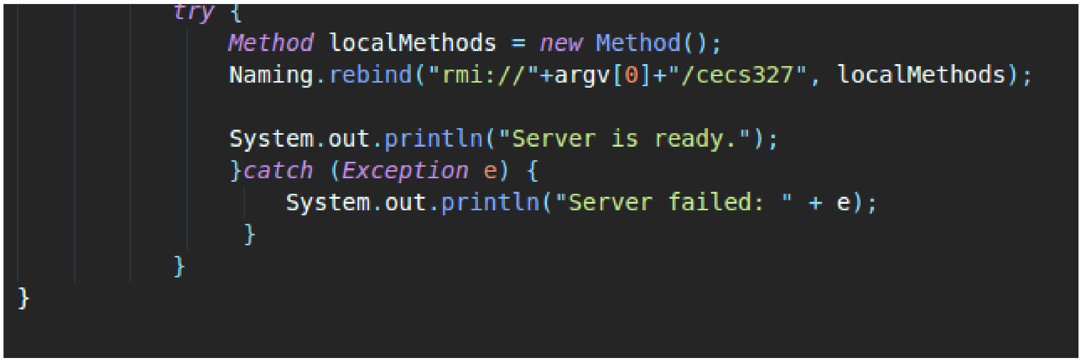
- Steal your own code

For **Assn6Server.java**

- This file you are building the handler for the RMI.
- It basically waits for a request and binds the Method to whatever was connecting to it. More details here: http://www.sce.carleton.ca/netmanage/simulator/rmi/RMIExplanation.htm



```
import java.rmi.*;
import java.rmi.server.*;

public class Assn6Server {
    public static void main (String[] argv) {
```

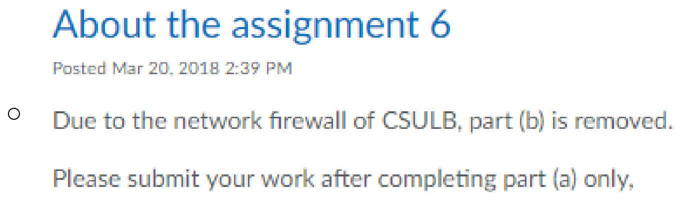


```
            try {
                Method localMethods = new Method();
                Naming.rebind("rmi://"+argv[0]+"/cecs327", localMethods);

                System.out.println("Server is ready.");
                }catch (Exception e) {
                    System.out.println("Server failed: " + e);
                 }
            }
}
```

- Most of us will have the same code for this.

Move / Compile

- ~~Move these three files into your server.~~



About the assignment 6

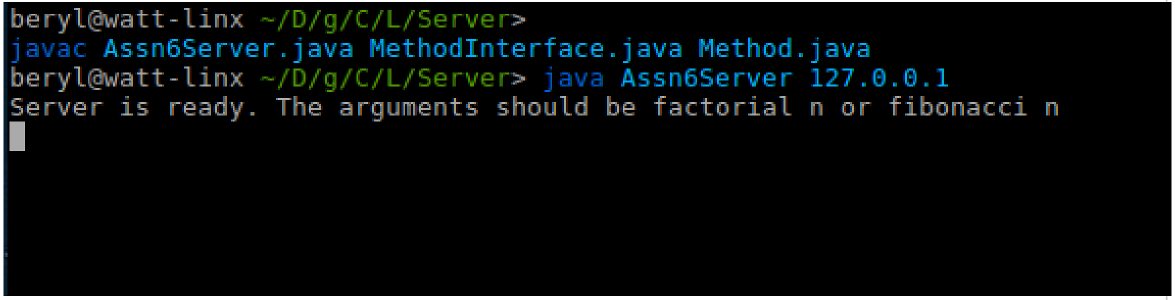
Posted Mar 20, 2018 2:39 PM

- Due to the network firewall of CSULB, part (b) is removed.

  Please submit your work after completing part (a) only.

local only

- Compile with: javac Assn6Server.java Method.java MethodInterface.java
- **RUN THIS BEFORE RUNNING Assn6Server**
  - `rmiregistry &`
  - You may have to nohup it as well to keep it running after you logout
  - `nohup rmiregistry &`
  - For Windows follow the Guide provided in the handout
    - WINDOWS INSTRUCTIONS
      - http://www.ejbtutorial.com/java-rmi/new-easy-tutorial-for-java-rmi-using-eclipse
      - Step 10 - 13, Note: This is using Eclipse
- Finally run the Assn6Server
  - `java Assn6Server`
- You should see the following:



```
beryl@watt-linx ~/D/g/C/L/Server>
javac Assn6Server.java MethodInterface.java Method.java
beryl@watt-linx ~/D/g/C/L/Server> java Assn6Server 127.0.0.1
Server is ready. The arguments should be factorial n or fibonacci n
```

- **DONE WITH SERVER SIDE**

CLIENT SIDE:

For the client side you need the two following files: Assn6Client.java and MethodInterface.java

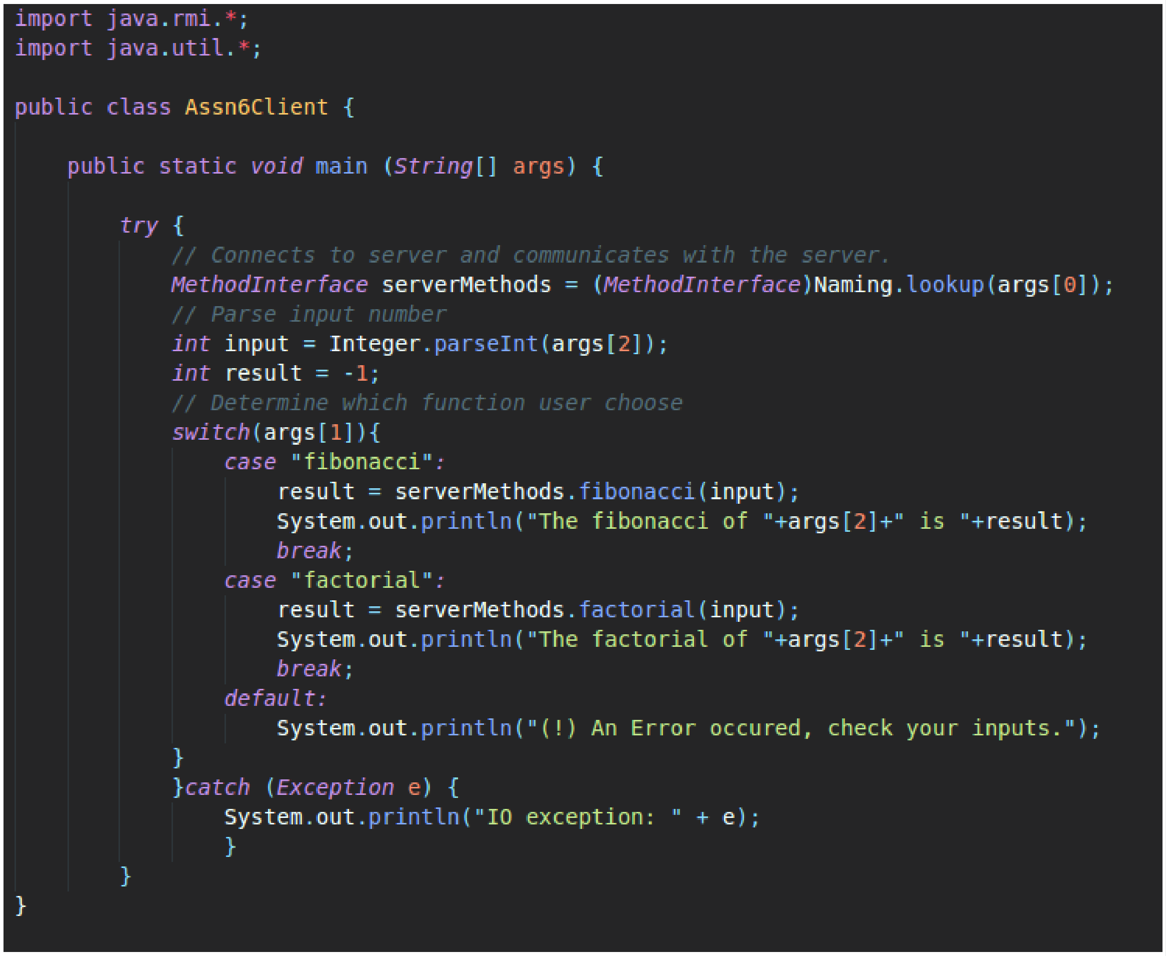
// Note: The Method Interface is the same for the server side. I'm not 100% sure we need this but his example he linked on the document used the same file. Will Confirm later on with him.

For **MethodInterface.java**

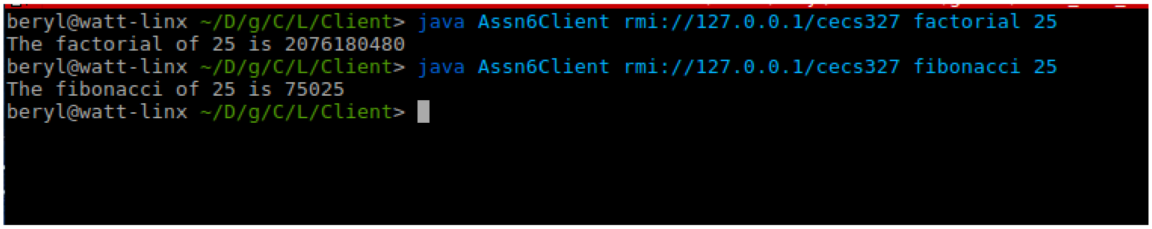
- Same as the Server side. Ctrl+C and Ctl+V.

For **Assn6Client.java**

- You need to take in three arguments
  - IP address where the RMI is stored
  - Which function you want to user
  - A number to be processed
- My Shit code

- 

```java
import java.rmi.*;
import java.util.*;

public class Assn6Client {

    public static void main (String[] args) {

        try {
            // Connects to server and communicates with the server.
            MethodInterface serverMethods = (MethodInterface)Naming.lookup(args[0]);
            // Parse input number
            int input = Integer.parseInt(args[2]);
            int result = -1;
            // Determine which function user choose
            switch(args[1]){
                case "fibonacci":
                    result = serverMethods.fibonacci(input);
                    System.out.println("The fibonacci of "+args[2]+" is "+result);
                    break;
                case "factorial":
                    result = serverMethods.factorial(input);
                    System.out.println("The factorial of "+args[2]+" is "+result);
                    break;
                default:
                    System.out.println("(!) An Error occured, check your inputs.");
            }
            }catch (Exception e) {
                System.out.println("IO exception: " + e);
                }
        }
}
```

- Compile and run
- **Note: this was executed locally with the server running in the background using the 127.0.0.1 IP address**
- 

```
beryl@watt-linx ~/D/g/C/L/Client> java Assn6Client rmi://127.0.0.1/cecs327 factorial 25
The factorial of 25 is 2076180480
beryl@watt-linx ~/D/g/C/L/Client> java Assn6Client rmi://127.0.0.1/cecs327 fibonacci 25
The fibonacci of 25 is 75025
beryl@watt-linx ~/D/g/C/L/Client>
```