University of Science and
Technology of Hanoi

Distributed System

Practical Work I

# TCP File Transfer

*Author:*
Nguyen Duc Dan
Nguyen Tri Huan
Nguyen Huu Chi Dat
Pham Minh Giang
Nguyen Xuan Duy Anh

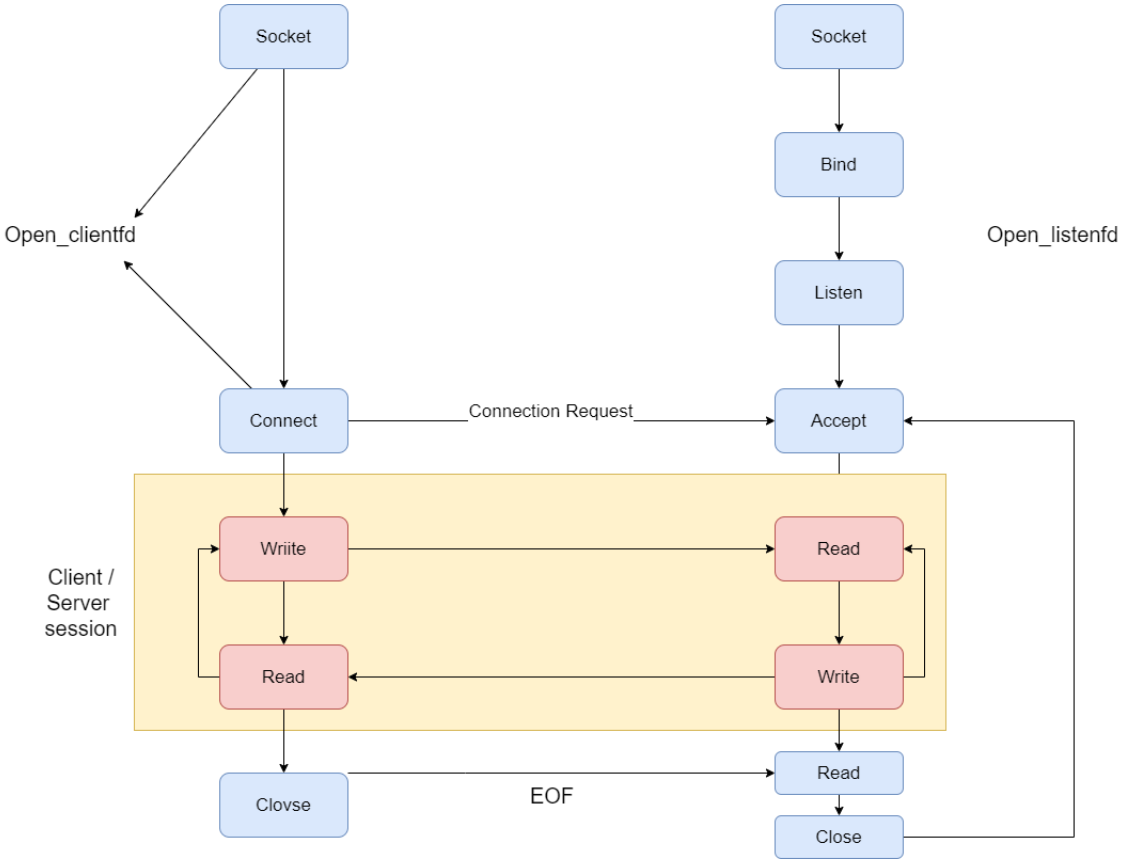
*Lecturer:*
Dr. Tran Giang Son

March 6, 2020

# Contents

# 1 Protocal Design

## 1.1 Figure




**Description** : Sockets help application process to communicate with each other using standard Unix file descriptors . Many routines exist to help ease the process of communication. Firstly, the server creates a socket, a communications endpoint. After that, we create a bind call that used to associate a socket with a port on the local machine, the port number is used by the kernel to match an incoming packet to a process. Then, we create listen call that used for waiting for incoming connections(we need to call bind() before you can listen). It's the end of open listenfd. Accept call gets the pending connection on the port you are listening on. The next step is the

Client/Server Session that help clients and servers communicate with each other by reading and writing to socket descriptors. In the final step, after closing Client/Server Session, we read END OF FILE on the server site to close the server connection

# 2 System Organizing

## 2.1 Open Session

- The server socket will be bound to port 4333 after created
- The server socket then listening to any message/data received

## 2.2 End open listen of server

- After getting the server socket to listening, the client socket will try to connect to the server

## 2.3 End open client socket

- In client/Server session, both client and server sending each other message alternatively until ones decided to close.

# 3 Code

## 3.1 Client

```
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.InputStream;
import java.net.Socket;

public class client {
public static void main(String[] args) throws
   IOException {
byte[] b = new byte[20002];
Socket sr = new Socket("localhost", 4333);
InputStream is = sr.getInputStream();
FileOutputStream fr = new
   FileOutputStream("labwork1/src/test_tcp_result.txt");
is.read(b, 0, b.length);
fr.write(b, 0, b.length);
                }
}
```

## 3.2 Server

```
import java.io.FileInputStream;
import java.io.IOException;
import java.io.OutputStream;
import java.net.ServerSocket;
import java.net.Socket;

public class server {
public static void main(String[] args) throws
   IOException {
ServerSocket s = new ServerSocket(4333);
Socket sr = s.accept();
FileInputStream fr = new
   FileInputStream("labwork1/src/test_tcp_input.txt");
byte[] b = new byte[20002];
```

```
fr.read(b, 0, b.length);
OutputStream os = sr.getOutputStream();
os.write(b, 0, b.length);
                }
}
```

# 4 Group participation

- Nguyen Duc Dan - BI8028 : Complete and implement the code
- Nguyen Xuan Duy Anh - BI8014 : System Organizing
- Nguyen Tri Huan - BI8069 : System Organizing
- Pham Minh Giang - BI8054 - : Draw Figure
- Nguyen Huu Chi Dat - BI8040: Summarize and Write the Report