Computer Network Lab

Exp 09: TCP Socket Programming

Dr. Ram P Rustagi
Sem V (2018-H2)
Dept of CSE, KSIT
rprustagi@ksit.edu.in

Ex10 Resources

- https://docs.oracle.com/javase/tutorial/networking/sockets/index.html
- https://www.geeksforgeeks.org/socket-programming-in-java/
- http://www.buyya.com/java/Chapter13.pdf
- Example: Java client server datagram programs
  - TCPClientTemplate.java
  - TCPClient.java (Later)
  - TCPServerTemplate.java
  - TCPServer.java (Later)
  - TCP Server variations: handling concurrent clients
    - TCP server: one thread for each client
    - TCP server using select.

Exp09 Description

- Program 09 (Java)
  - Using TCP/IP sockets, write a client – server program to make the client send the file name and to make the server send back the contents of the requested file if present

Lab Program: TCP Server

- Lab program expectation: Server program
  - A TCP Server program that will communicate with many clients. It should act like a chat server.
  - Port number is taken from command line argument
    - IP is taken by default (all)
  - It receives a filename (with path) from client.
  - Server reads the file content from the file and sends it to the requesting client.
  - Server forever waits for a client request
  - The program runs for ever.
  - All method calls should be checked for errors

Lab Program: TCP Client

- Lab program expectation: Client program
  - A TCP client program that will communicate with a TCP Server
  - The input command line parameters are
    - Server IP Address
    - Server Port number
    - List of filename(s)
  - Client sends the filename(s) to server on specified port, one at a time.
  - It should wait for the response (file content) from the server. If file doesn't exist, server sends "-1".
  - It should write received content into the file.
  - All methods/system calls to be checked for errors

TCP Data Concepts

- Stream packet
  - Data is delivered in stream of bytes
  - Each client connection is identified by a new socket (file handle).
  - Doesn't need server or client address after the connection is made.
- Key Java classes for sending files over TCP
  - `OutputStream`
  - `InputStream`
  - `BufferedReader`
  - `Socket`
  - `ServerSocket`
  - `File`
  - `PrintWriter`

TCP Socket

- Java class constructors
  - `https://docs.oracle.com/javase/8/docs/technotes/guides/net/index.html`
  - Socket for client
    - `sock = new Socket(serverIPAddress, serverPort)`
  - Socket for server (binds to specified port)
    - `ssock = new ServerSocket(serverport, backlog)`
- Error / exception
  - `SocketException`

Stream Socket Methods

- To send the filename as TCP data

```
ostream=sock.getOutputStream();
printwriter=new PrintWriter(ostream,true);
printwriter.println(filename);
```

- To receive data by server

```
csock = ssock.accept()
clientIP =
csock.getInetAddress().getHostAddress();
clientPort = csock.getPort();
istream = csock.getInputStream( );
sockReader = new BufferedReader(new
InputStreamReader(istream));
filename = sockReader.readLine();
```

TCP Server : Reading from File

- Checking if file exists

```
tmpfile = new File(filename);
if (! tmpfile.exists() ) {
   pwrite.println(-1);
   csock.close();
}
```

- Reading file content and sending to client

```
ostream = csock.getOutputStream();
PrintWriter pwrite=new PrintWriter(ostream,
true);
fileReader = new BufferedReader(new
FileReader(filename));
String str;
while((str=fileReader.readLine())!=null){
    pwrite.println(str);
}
```

TCP Client & Server Invocation

- `java tcpClient 10.26.30.11 2345 f1 f2 …`
  - arg[0] : `10.26.30.11` (Server IP Address)
  - arg[1]: `2345` (Server port number)
  - arg[2] to arg[n]: list of files
- `java tcpServer 2345`
  - arg[0]: `2345` (Server port number)

- Use nc to test TCP Server
  - `nc 10.26.30.11 2345 file1.txt`
  - it should display content of file or (-1 if file does not exist)

Basic TCP Server and Client

- `nc -l 3333`
  - Runs an TCP Server on port `3333`
- `nc 10.26.30.11 3333`
  - Run an TCP Client and connect to a TCP server running on `10.26.33.11` on port `3333`.
- Running java program
  - `javac TCPClient.java`
  - `javac TCPServer.java`
  - `java -classpath .  TCPClient`
  - `java -classpath .  TCPServer`

Template Programs

- Template programs
- TCPClientTemplate.java
- TCPServerTemplate.java
- How to use Template programs
  - Copy them to TCPClient.java, TCPServer.java
  - Fill in the lines with ??
  - Do all the error checking and throw exceptions
    - The sample program template does not have any error checking.
  - Run the program and test the server with multiple clients.

Program Expectation.

- The TCP server program should run for ever.
- Your TCP client and server should be able to work with nc
- Your TCP client should be able to work with your TCP Server.
- Your TCP Client should be able to work with any one's TCP server and vice versa,
- Your program should not crash with any bad user input.
- TCP Server program should be able to work with multiple clients concurrently.
- Do all required validation and check all exceptions.