Network Architecture 1

SPRING 2016 PROJECT 1

Submission

BY

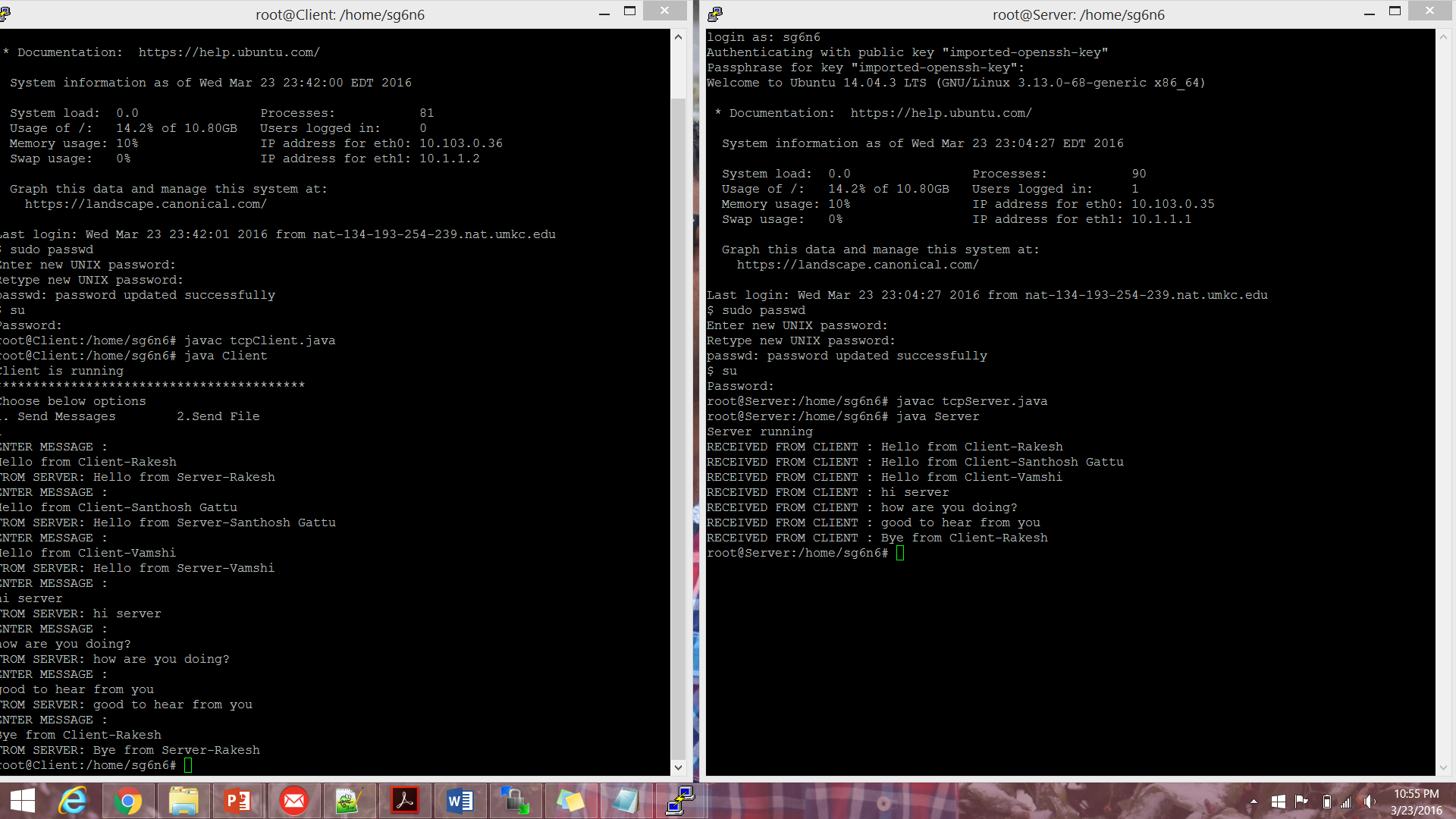
Santhosh Kumar Gattu (sg6n6)

**Part I. GENI/Socket programming Warm-up (50%)**

**Develop and deploy a simple TCP client and server programs on GENI (refer GENI LabZero slide and sample socket programs). Show the screenshots of simple message exchanges.**

1. **(20%) Start from client message ‘Hello from Client-your names’ and server responses with ‘Hello from Server-your names’. Then messages from each side are echoed to each other. The program quit the program with typing ‘Bye from Client-your name’ and ‘Bye from Server-your name’.**





**Server Program in Java programming language**

import java.io.\*;

import java.net.\*;

class Server {

public static void main(String argv[]) throws Exception

{

String clientSentence;

String returnSentence="";

System.out.println("Server running");

ServerSocket welcomeSocket = new ServerSocket(4567);

while(true) {

Socket connectionSocket = welcomeSocket.accept();

BufferedReader inFromClient =

new BufferedReader(new

InputStreamReader(connectionSocket.getInputStream()));

DataOutputStream outToClient =

new DataOutputStream(connectionSocket.getOutputStream());

clientSentence = inFromClient.readLine();

System.out.println("RECEIVED FROM CLIENT : "+clientSentence);

if(clientSentence.contains("Hello from Client")){

String[] parts = clientSentence.split("-");

returnSentence = "Hello from Server-"+parts[1]+ '\n';

}

else if(clientSentence.startsWith("Bye from Client")){

String[] parts = clientSentence.split("-");

returnSentence = "Bye from Server-"+parts[1]+ '\n';

outToClient.writeBytes(returnSentence);

break;

}

else{

returnSentence = clientSentence+ '\n';

}

outToClient.writeBytes(returnSentence);

}

}

}

**Client Program in Java programming language**

import java.io.\*;

import java.net.\*;

import java.util.Scanner;

class Client {

public static void main(String argv[]) throws Exception

{

String sentence = "";

String serverSentence="";

BufferedReader inFromUser =

new BufferedReader(new InputStreamReader(System.in));

System.out.println("Client is running");

System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

System.out.println("Choose below options");

System.out.println("1. Send Messages\t2.Send File");

Scanner scanner = new Scanner(System.in);

int option = scanner.nextInt();

if(option==1){

//opening socket connection

Socket clientSocket=null;

while(!serverSentence.startsWith("Bye from Server")){

System.out.println("ENTER MESSAGE : ");

clientSocket= new Socket("152.54.14.39", 4567);

DataOutputStream outToServer =

new DataOutputStream(clientSocket.getOutputStream());

BufferedReader inFromServer =

new BufferedReader(new

InputStreamReader(clientSocket.getInputStream()));

sentence = inFromUser.readLine();

outToServer.writeBytes(sentence + '\n');

serverSentence = inFromServer.readLine();

System.out.println("FROM SERVER: " + serverSentence);

}

clientSocket.close();

}

else if(option==2){

System.out.println("Choose file");

}

else{

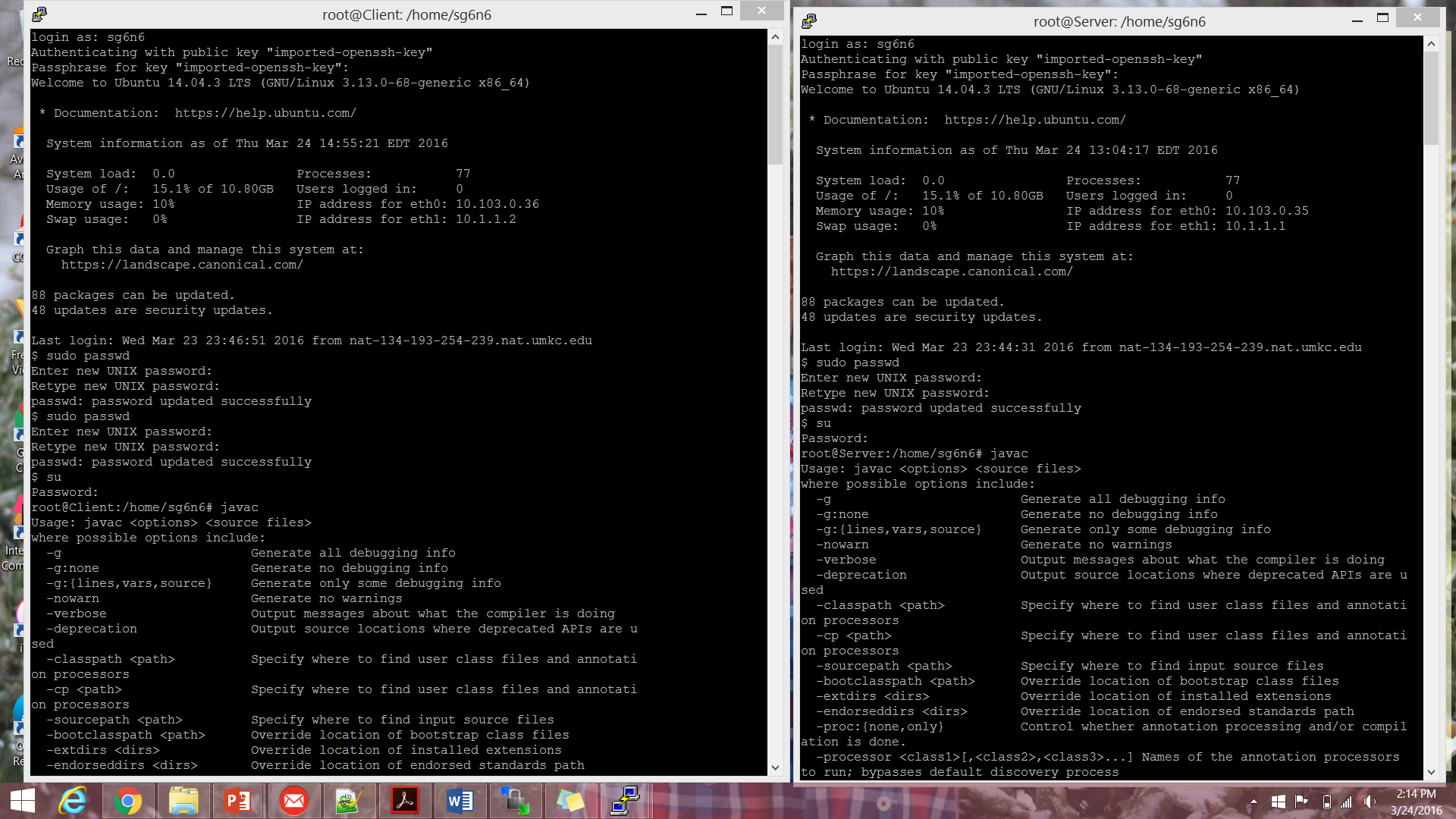
System.out.println("Choose correct option");

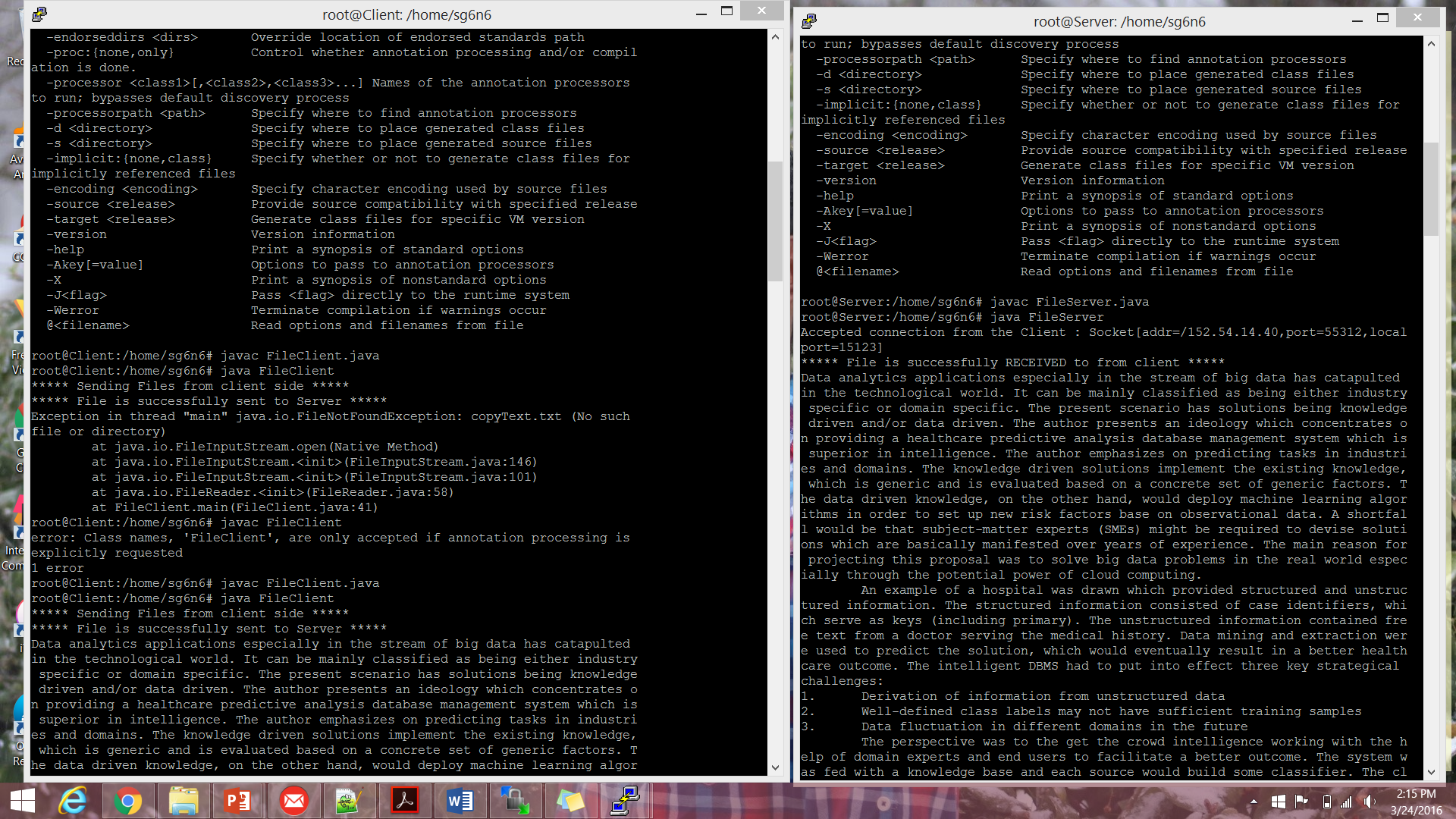
}

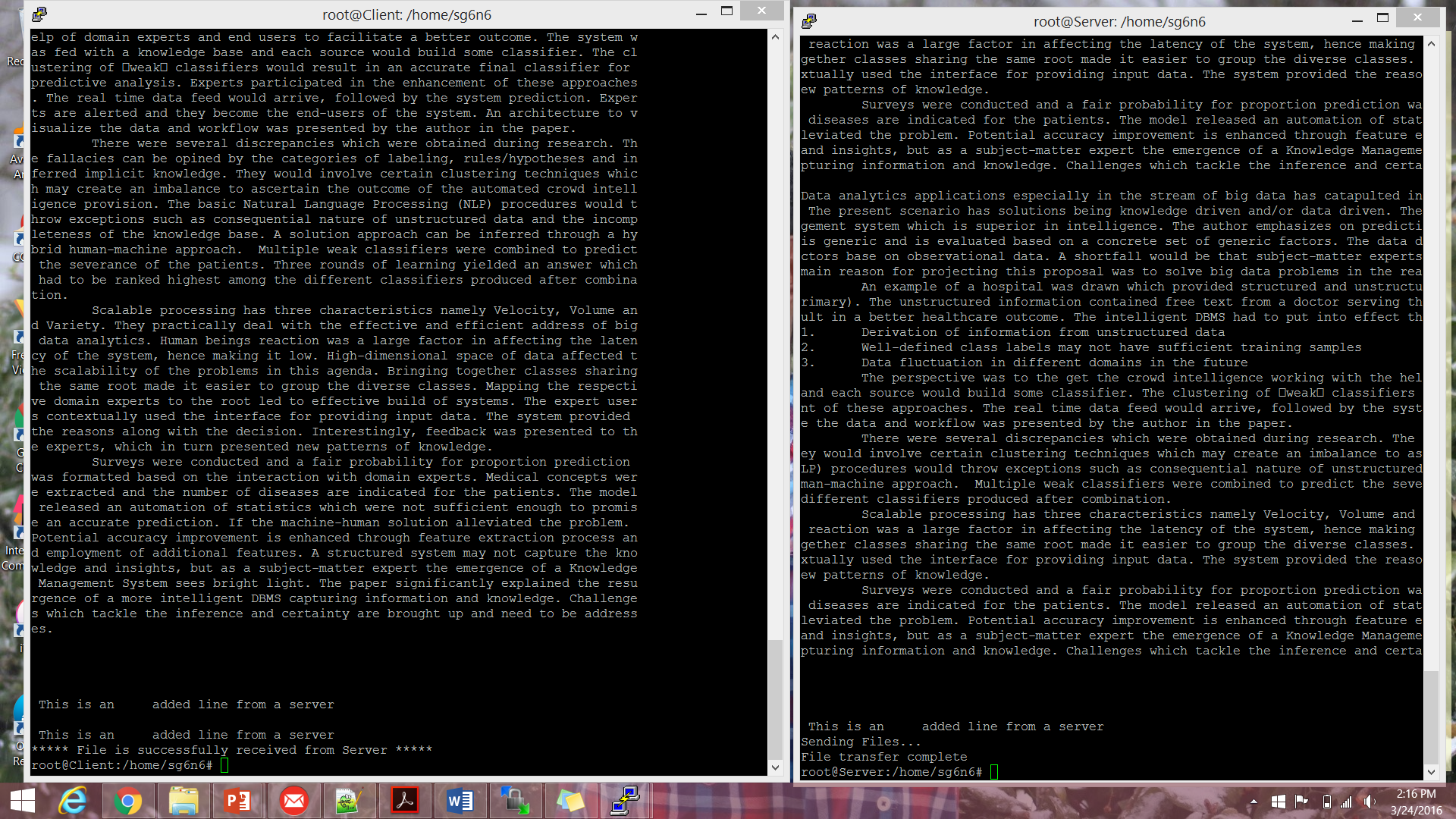
}

}

1. **(30%) A client sends a text file (> 10KB) to a server. Server prints the file on the screen, Server saves the file in a local system, Server appends one more line (eg. ‘This is an added line from a server) to the file, And send the updated file back to the client. Client shows the file on the screen after it fully receives the file.**







**Client:**

**import java.net.\*;**

**import java.io.\*;**

**class FileClient {**

**public static void main (String [] args ) throws IOException {**

**int filesize=2000000;**

**int bytesRead;**

**String line = null;**

**int currentTot = 0;**

**Socket socket = new Socket("152.54.14.39",15123);**

**File filetobesent = new File("Text.txt");**

**byte [] bytearray = new byte [(int)filetobesent.length()];**

**FileInputStream fin = new FileInputStream(filetobesent);**

**BufferedInputStream bin = new BufferedInputStream(fin);**

**bin.read(bytearray,0,bytearray.length);**

**OutputStream os = socket.getOutputStream();**

**System.out.println("\*\*\*\*\* Sending Files from client side \*\*\*\*\*");**

**os.write(bytearray,0,bytearray.length);**

**os.flush();**

**System.out.println("\*\*\*\*\* File is successfully sent to Server \*\*\*\*\*");**

**bytearray = new byte [filesize];**

**bytearray = new byte [filesize];**

**InputStream is = socket.getInputStream();**

**FileOutputStream fos = new FileOutputStream("Text.txt");**

**BufferedOutputStream bos = new BufferedOutputStream(fos);**

**bytesRead = is.read(bytearray,0,bytearray.length);**

**currentTot = bytesRead;**

**do {**

**bytesRead =**

**is.read(bytearray, currentTot, (bytearray.length-currentTot));**

**if(bytesRead >= 0) currentTot += bytesRead;**

**} while(bytesRead > -1);**

**bos.write(bytearray, 0 , currentTot);**

**bos.flush();**

**bos.close();**

**socket.close();**

**BufferedReader br = new BufferedReader(new FileReader("Text.txt"));**

**while ((line = br.readLine()) != null) {**

**System.out.println(line);**

**}**

**System.out.println("\*\*\*\*\* File is successfully received from Server \*\*\*\*\*");**

**}**

**}**

**Server:**

**import java.net.\*;**

**import java.io.\*;**

**public class FileServer {**

**public static void main (String [] args ) throws IOException {**

**int filesize=2000000;**

**int bytesRead=0;**

**int currentTot = 0;**

**String line = null;**

**ServerSocket serverSocket = new ServerSocket(15123);**

**Socket socket = serverSocket.accept();**

**System.out.println("Accepted connection from the Client : " + socket);**

**byte [] bytearray = new byte [filesize];**

**InputStream is = socket.getInputStream();**

**FileOutputStream fos = new FileOutputStream("copyText.txt");**

**BufferedOutputStream bos = new BufferedOutputStream(fos);**

**bytesRead = is.read(bytearray,0,bytearray.length);**

**currentTot = bytesRead;**

**/\*do {**

**bytesRead =**

**is.read(bytearray, currentTot, (bytearray.length-currentTot));**

**if(bytesRead >= 0) currentTot += bytesRead;**

**} while(bytesRead > -1);\*/**

**bos.write(bytearray, 0 , currentTot);**

**bos.flush();**

**bos.close();**

**System.out.println("\*\*\*\*\* File is successfully RECEIVED to from client \*\*\*\*\*");**

**BufferedReader br = new BufferedReader(new FileReader("copyText.txt"));**

**while ((line = br.readLine()) != null) {**

**System.out.println(line);**

**}**

**FileWriter fw = new FileWriter("copyText.txt" ,true);**

**fw.append("\n\n This is an added line from a server");**

**fw.close();**

**File transferFile = new File ("copyText.txt");**

**//append(transferFile, "\n\n This is an added line from a server");**

**bytearray = new byte [(int)transferFile.length()];**

**FileInputStream fin = new FileInputStream(transferFile);**

**BufferedInputStream bin = new BufferedInputStream(fin);**

**bin.read(bytearray,0,bytearray.length);**

**OutputStream os = socket.getOutputStream();**

**System.out.println("Sending Files...");**

**os.write(bytearray,0,bytearray.length);**

**os.flush();**

**socket.close();**

**System.out.println("File transfer complete");**

**}**

**public static void append(File aFile, String content) {**

**FileWriter fw;**

**try {**

**fw = new FileWriter(aFile.getAbsoluteFile());**

**BufferedWriter bw = new BufferedWriter(fw);**

**fw.append(content);**

**fw.close();**

**System.out.println("Done");**

**} catch (IOException e) {**

**// TODO Auto-generated catch block**

**e.printStackTrace();**

**}**

**}**

**}**

**Part-2**

**a.**



**Client**

**import java.io.\*;**

**import java.net.\*;**

**import java.util.Scanner;**

**class Client2a {**

**public static void main(String argv[]) throws Exception**

**{**

**String sentence = "";**

**String serverSentence="";**

**BufferedReader inFromUser =**

**new BufferedReader(new InputStreamReader(System.in));**

**System.out.println("Client is running");**

**System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");**

**//opening socket connection**

**Socket clientSocket=null;**

**while(!sentence.equals("exit")){**

**System.out.println("ENTER MESSAGE : ");**

**clientSocket= new Socket("152.54.14.22", 4567);**

**DataOutputStream outToServer =**

**new DataOutputStream(clientSocket.getOutputStream());**

**BufferedReader inFromServer =**

**new BufferedReader(new**

**InputStreamReader(clientSocket.getInputStream()));**

**sentence = inFromUser.readLine();**

**outToServer.writeBytes(sentence + '\n');**

**serverSentence = inFromServer.readLine();**

**System.out.println("FROM SERVER: " + serverSentence);**

**}**

**clientSocket.close();**

**}**

**}**

**Server**

**import java.io.\*;**

**import java.net.\*;**

**class Server2a {**

**public static void main(String argv[]) throws Exception**

**{**

**BufferedReader inFromUser =**

**new BufferedReader(new InputStreamReader(System.in));**

**String clientSentence;**

**String returnSentence="";**

**System.out.println("Server running");**

**ServerSocket welcomeSocket = new ServerSocket(37946);**

**String sentence;**

**while(true) {**

**Socket connectionSocket = welcomeSocket.accept();**

**BufferedReader inFromClient =**

**new BufferedReader(new**

**InputStreamReader(connectionSocket.getInputStream()));**

**DataOutputStream outToClient =**

**new DataOutputStream(connectionSocket.getOutputStream());**

**clientSentence = inFromClient.readLine();**

**System.out.println("RECEIVED FROM CLIENT : "+clientSentence);**

**if(clientSentence.equals("exit")){**

**sentence="exit";**

**returnSentence = sentence+ '\n';**

**outToClient.writeBytes(returnSentence);**

**connectionSocket.close();**

**welcomeSocket.close();**

**return;**

**}**

**else{**

**sentence = inFromUser.readLine();**

**returnSentence = sentence+ '\n';**

**outToClient.writeBytes(returnSentence);**

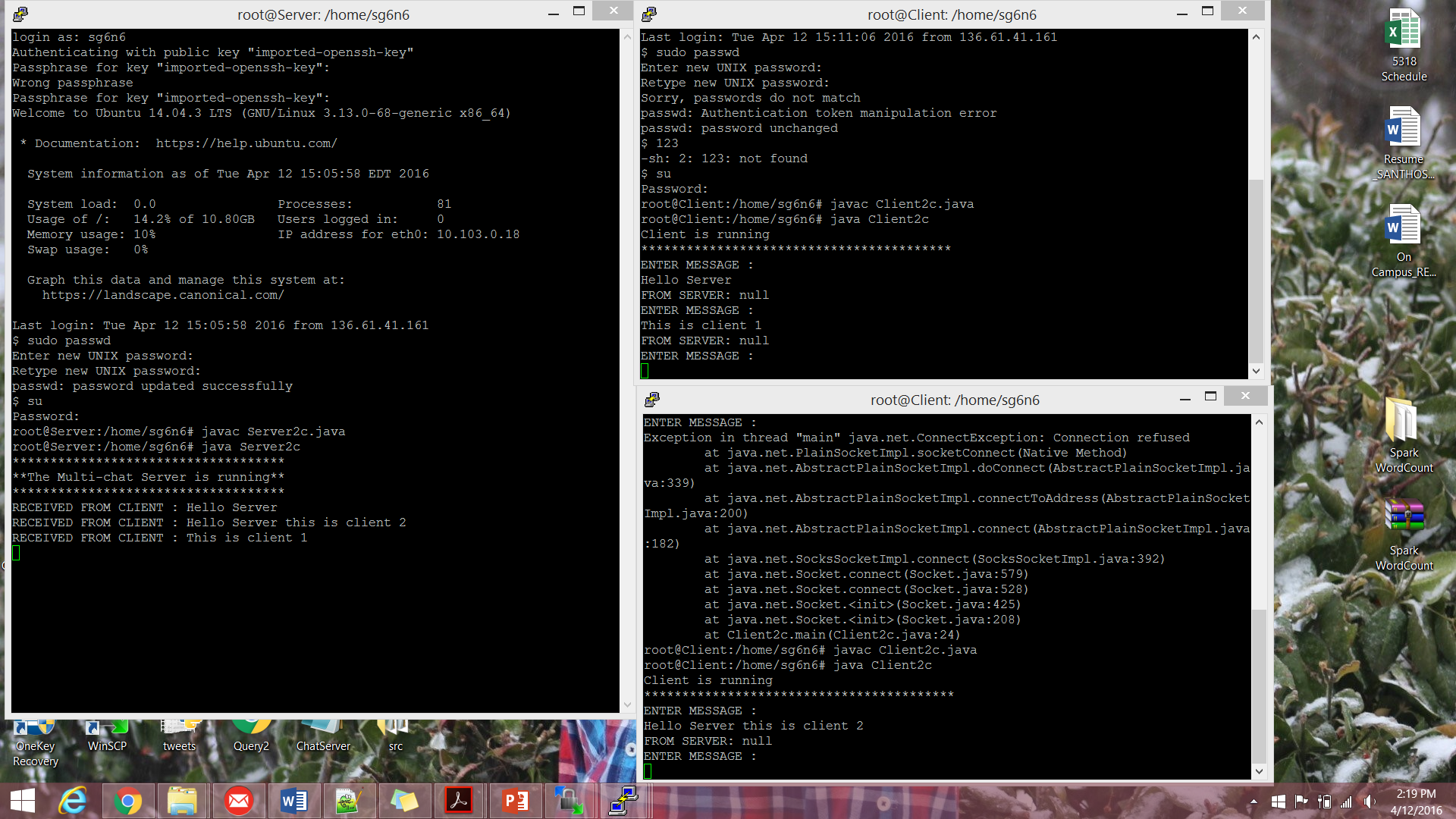
**}**

**}**

**}**

**}**

**2c.**



**Client**

**import java.io.\*;**

**import java.net.\*;**

**import java.util.Scanner;**

**class Client2c {**

**public static void main(String argv[]) throws Exception**

**{**

**String sentence = "";**

**String serverSentence="";**

**BufferedReader inFromUser =**

**new BufferedReader(new InputStreamReader(System.in));**

**System.out.println("Client is running");**

**System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");**

**//opening socket connection**

**Socket clientSocket=null;**

**while(!sentence.equals("exit")){**

**System.out.println("ENTER MESSAGE : ");**

**clientSocket= new Socket("152.54.14.22", 4567);**

**DataOutputStream outToServer =**

**new DataOutputStream(clientSocket.getOutputStream());**

**BufferedReader inFromServer =**

**new BufferedReader(new**

**InputStreamReader(clientSocket.getInputStream()));**

**sentence = inFromUser.readLine();**

**outToServer.writeBytes(sentence + '\n');**

**serverSentence = inFromServer.readLine();**

**System.out.println("FROM SERVER: " + serverSentence);**

**}**

**clientSocket.close();**

**}**

**}**

**Server**

**import java.io.BufferedReader;**

**import java.io.IOException;**

**import java.io.InputStreamReader;**

**import java.io.PrintWriter;**

**import java.net.ServerSocket;**

**import java.net.Socket;**

**public class Server2c {**

**public static void main(String[] args) throws Exception {**

**System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");**

**System.out.println("\*\*The Multi-chat Server is running\*\*");**

**System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");**

**ServerSocket listener = new ServerSocket(9987);**

**try {**

**while (true) {**

**new Handler(listener.accept()).start();**

**}**

**} finally {**

**listener.close();**

**}**

**}**

**private static class Handler extends Thread {**

**private String name;**

**private Socket socket;**

**private BufferedReader in;**

**private PrintWriter out;**

**static int count=1;**

**private String Clientname="client"+count;**

**public Handler(Socket socket) {**

**this.socket = socket;**

**count++;**

**}**

**public void run() {**

**try {**

**in = new BufferedReader(new InputStreamReader(**

**socket.getInputStream()));**

**out = new PrintWriter(socket.getOutputStream(), true);**

**/\*for (PrintWriter writer : writers) {**

**writer.println("MESSAGE " + name + ": " + input);**

**}\*/**

**String msg = in.readLine();**

**System.out.println("RECEIVED FROM CLIENT : "+msg);**

**} catch (IOException e) {**

**System.out.println(e);**

**} finally {**

**try {**

**socket.close();**

**}**

**catch (IOException e) {**

**}**

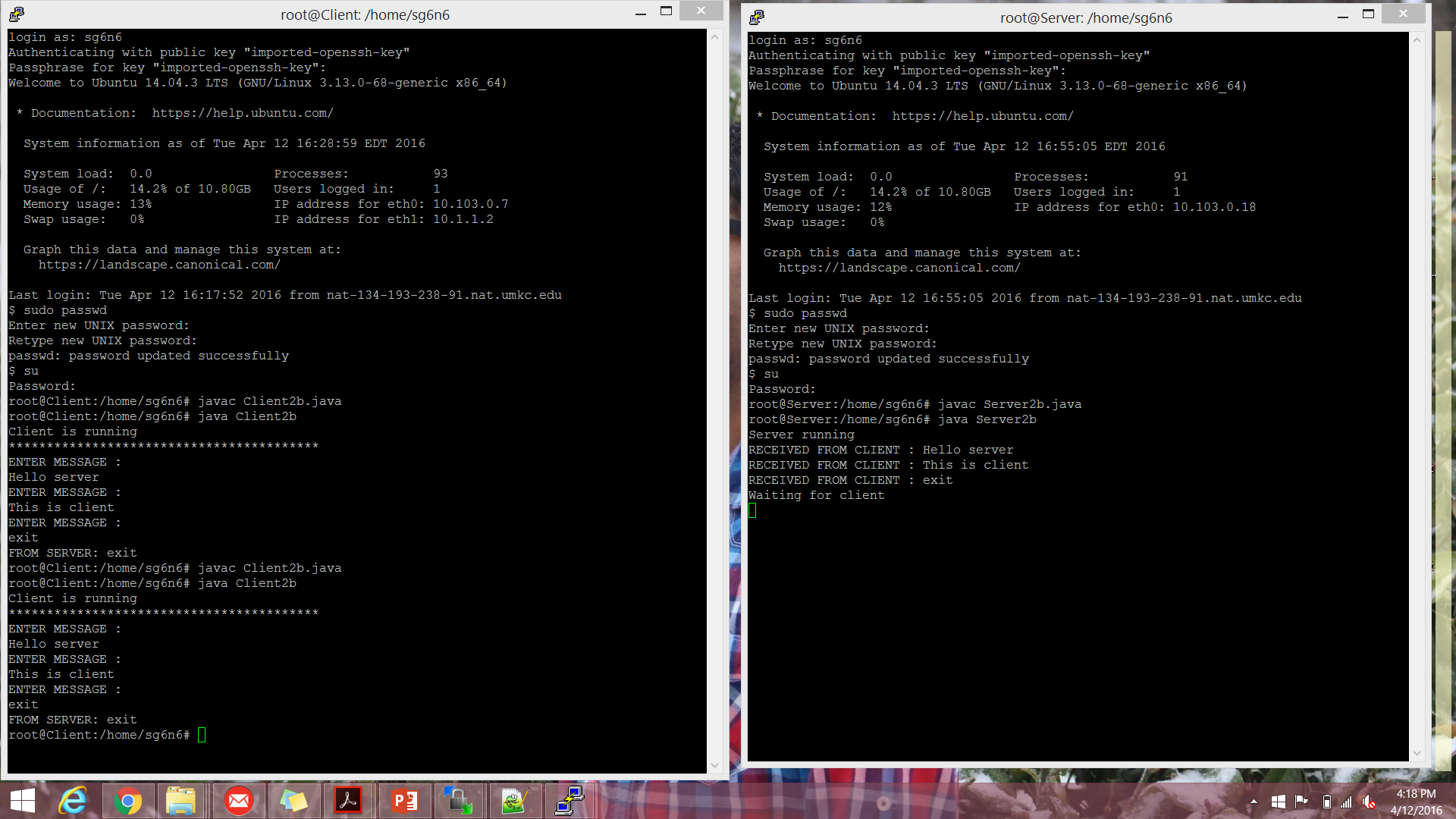
**}**

**}**

**}**

**}**

**2.b**



**Client**

**import java.io.\*;**

**import java.net.\*;**

**import java.util.Scanner;**

**class Client2b {**

**public static void main(String argv[]) throws Exception**

**{**

**String sentence = "";**

**String serverSentence="";**

**BufferedReader inFromUser =**

**new BufferedReader(new InputStreamReader(System.in));**

**System.out.println("Client is running");**

**System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");**

**//opening socket connection**

**Socket clientSocket=null;**

**while(!sentence.equals("exit")){**

**System.out.println("ENTER MESSAGE : ");**

**clientSocket= new Socket("152.54.14.22", 1234);**

**DataOutputStream outToServer =**

**new DataOutputStream(clientSocket.getOutputStream());**

**BufferedReader inFromServer =**

**new BufferedReader(new**

**InputStreamReader(clientSocket.getInputStream()));**

**sentence = inFromUser.readLine();**

**outToServer.writeBytes(sentence + '\n');**

**if(sentence.equals("exit")){**

**serverSentence = inFromServer.readLine();**

**System.out.println("FROM SERVER: " + serverSentence);**

**}**

**}**

**clientSocket.close();**

**}**

**}**

**Server**

**import java.io.\*;**

**import java.net.\*;**

**class Server2b {**

**public static void main(String argv[]) throws Exception**

**{**

**int flag=0;**

**BufferedReader inFromUser =**

**new BufferedReader(new InputStreamReader(System.in));**

**String clientSentence="";**

**String returnSentence="";**

**System.out.println("Server running");**

**//ServerSocket welcomeSocket = new ServerSocket(4567);**

**while(true){**

**String sentence;**

**ServerSocket welcomeSocket = new ServerSocket(1234);**

**while(true&flag==0) {**

**Socket connectionSocket = welcomeSocket.accept();**

**BufferedReader inFromClient =**

**new BufferedReader(new**

**InputStreamReader(connectionSocket.getInputStream()));**

**DataOutputStream outToClient =**

**new DataOutputStream(connectionSocket.getOutputStream());**

**clientSentence = inFromClient.readLine();**

**System.out.println("RECEIVED FROM CLIENT : "+clientSentence);**

**if(clientSentence.equals("exit")){**

**sentence="exit";**

**returnSentence = sentence+ '\n';**

**System.out.println("Waiting for client");**

**outToClient.writeBytes(returnSentence);**

**outToClient.close();**

**connectionSocket.close();**

**welcomeSocket.close();**

**flag=1;**

**}**

**else{**

**//sentence = inFromUser.readLine();**

**//returnSentence = sentence+ '\n';**

**//outToClient.writeBytes(returnSentence);**

**outToClient.close();**

**}**

**}**

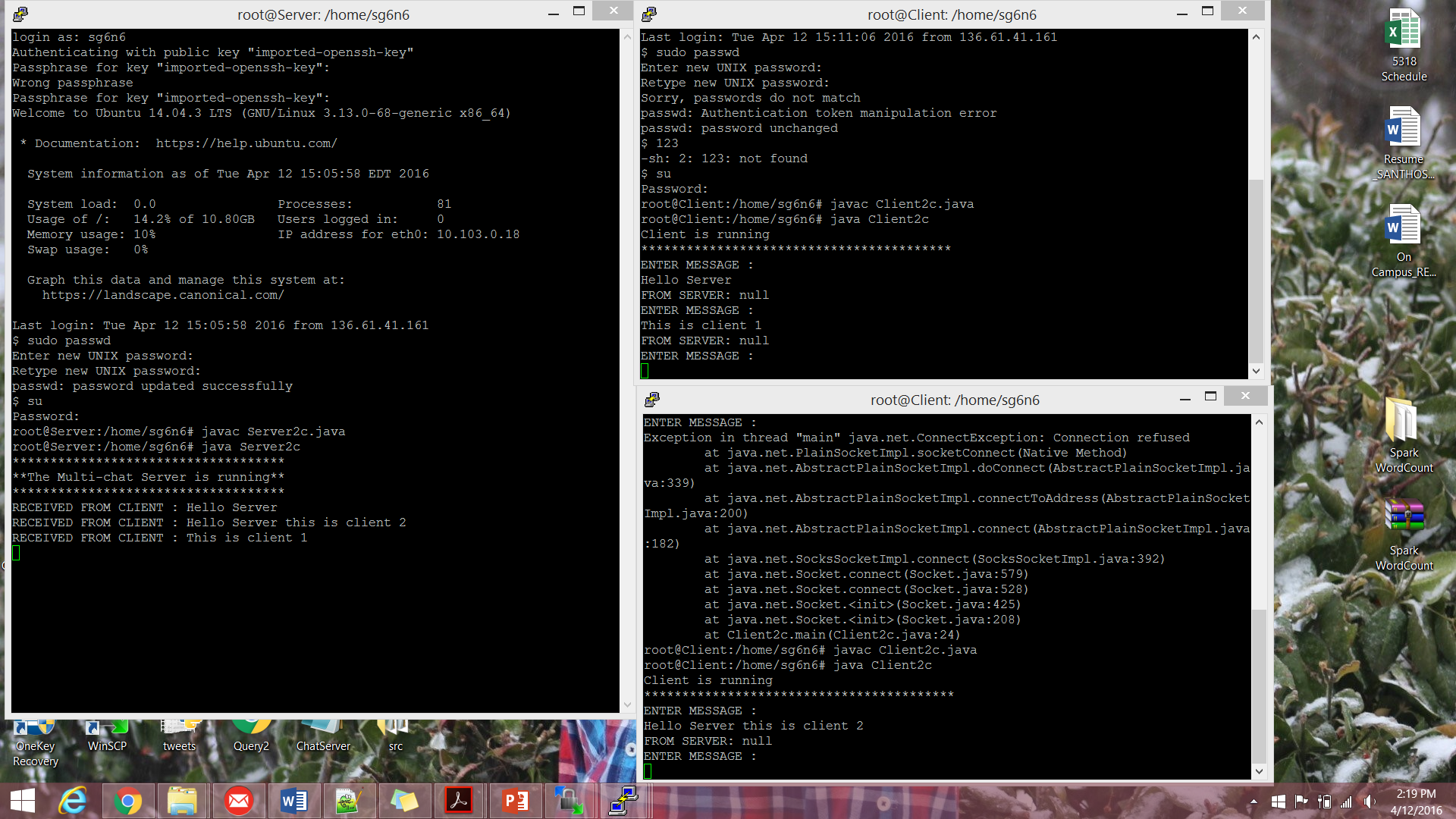
**flag=0;**

**}**

**}**

**}**

**2.d**



**Server**

**import java.io.BufferedReader;**

**import java.io.IOException;**

**import java.io.InputStreamReader;**

**import java.io.PrintWriter;**

**import java.net.ServerSocket;**

**import java.net.Socket;**

**import java.util.HashSet;**

**public class Server2d {**

**private static HashSet<PrintWriter> writers = new HashSet<PrintWriter>();**

**public static void main(String[] args) throws Exception {**

**System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");**

**System.out.println("\*\*The Multi-chat Server is running\*\*");**

**System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");**

**int i=0;**

**ServerSocket serverSocket = new ServerSocket(6789);**

**try {**

**while (true) {**

**new ClientManager(serverSocket.accept(),"Client1"+i).start();**

**i++;**

**}**

**} finally {**

**serverSocket.close();**

**}**

**}**

**private static class ClientManager extends Thread {**

**private Socket socket;**

**private BufferedReader in;**

**private PrintWriter out;**

**private String Clientname;**

**public ClientManager(Socket socket,String name) {**

**this.socket = socket;**

**Clientname=name;**

**}**

**public void run() {**

**try {**

**in = new BufferedReader(new InputStreamReader(**

**socket.getInputStream()));**

**out = new PrintWriter(socket.getOutputStream(), true);**

**writers.add(out);**

**String msg = in.readLine();**

**System.out.println("RECEIVED FROM CLIENT : "+ Clientname+" and Message: "+msg);**

**for (PrintWriter writer : writers) {**

**writer.println("MESSAGE from all clients " + msg);**

**}**

**} catch (IOException e) {**

**System.out.println(e);**

**} finally {**

**try {**

**socket.close();**

**}**

**catch (IOException e) {**

**}**

**}**

**}**

**}**

**}**

**Client**

**import java.io.\*;**

**import java.net.\*;**

**import java.util.Scanner;**

**class Client2d {**

**BufferedReader in;**

**//PrintWriter out;**

**public static void main(String argv[]) throws Exception**

**{**

**String sentence = "";**

**String serverSentence="";**

**PrintWriter out;**

**BufferedReader inFromUser =**

**new BufferedReader(new InputStreamReader(System.in));**

**System.out.println("Client is running");**

**System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");**

**//opening socket connection**

**Socket clientSocket=null;**

**int flag=0;**

**/\*clientSocket= new Socket("127.0.0.1", 4567);**

**try {**

**while (true) {**

**new ClientThread(clientSocket).start();**

**}**

**} finally {**

**clientSocket.close();**

**}\*/**

**while(!sentence.equals("exit")){**

**System.out.println("ENTER MESSAGE : ");**

**clientSocket= new Socket("152.54.14.22", 6789);**

**DataOutputStream outToServer =**

**new DataOutputStream(clientSocket.getOutputStream());**

**BufferedReader inFromServer =**

**new BufferedReader(new**

**InputStreamReader(clientSocket.getInputStream()));**

**out = new PrintWriter(clientSocket.getOutputStream(), true);**

**sentence = inFromUser.readLine();**

**outToServer.writeBytes(sentence + '\n');**

**serverSentence = inFromServer.readLine();**

**System.out.println("FROM SERVER: " + serverSentence);**

**}**

**clientSocket.close();**

**}**

**private static class ClientThread extends Thread {**

**private Socket socket;**

**private BufferedReader in;**

**private PrintWriter out;**

**public ClientThread(Socket socket) {**

**this.socket = socket;**

**}**

**public void run() {**

**try {**

**in = new BufferedReader(new InputStreamReader(**

**socket.getInputStream()));**

**out = new PrintWriter(socket.getOutputStream(), true);**

**String msg = in.readLine();**

**System.out.println("RECEIVED FROM Server and Message: "+msg);**

**} catch (IOException e) {**

**System.out.println(e);**

**} finally {**

**try {**

**socket.close();**

**}**

**catch (IOException e) {**

**}**

**}**

**}**

**}\*/**

**}**