Vivek Patel

CSC435

InetServer

InetServer code + comments

import java.io.\*;

import java.net.\*;

class Worker extends Thread {

Socket sock;

Worker (Socket s) {sock = s;}

public void run() { // runs the server functions, including input and output as well as features (in this case IP lookup)

PrintStream out = null;

BufferedReader in = null;

try { // tries to create an input reader and an output reader

in = new BufferedReader(new InputStreamReader (sock.getInputStream()));

out = new PrintStream(sock.getOutputStream());

try { // tries to read the input from the client, process it, and print the outout to the client

String name;

name = in.readLine(); // takes the input

System.out.println("Looking up " + name);

printRemoteAddress(name, out);

} catch (IOException x) { // catches an IO exception and prints an error message

System.out.println("Server read error");

x.printStackTrace();

}

sock.close(); // closes the connection to this specific client

} catch (IOException ioe) {System.out.println(ioe);} // catches an IO exception and prints an error message

}

static void printRemoteAddress (String name, PrintStream out){ // looks up the IP address and Host name of a website or IP address given to it

//and prints it to the output stream given to it

try {

out.println("Looking up " + name + "...");

InetAddress machine = InetAddress.getByName(name); // looks up the site/IP address with the name that was given

out.println("Host name: " + machine.getHostName()); // looks up the host name and prints it

out.println("Host IP: " + toText(machine.getAddress())); // looks up the IP address and prints it

} catch(UnknownHostException ex) { // catches an error when looking up a website and prints an error message

out.println("Failed in attempt to look up " + name);

}

}

static String toText(byte ip[]){ //takes an IP address and returns it as a String

StringBuffer result = new StringBuffer();

for (int i = 0; i < ip.length; i++){

if (i > 0) result.append(".");

result.append (0xff & ip[i]);

}

return result.toString();

}

}

public class InetServer{

public static void main(String[] args) throws IOException {//Starts the server and deploys workers as needed

int q\_len = 6;

int port = 1565; // the port connection the server is listening to

Socket sock;

ServerSocket servsock = new ServerSocket(port, q\_len); //initializes the server

System.out.println ("Clark Elliot's InetServer 1.8 starting up, listening at port 1565.\n");

while (true) {

sock = servsock.accept(); // listens for a client

new Worker(sock).start(); // creates a worker for the client

}

}

}

InetClient code + comments

import java.io.\*;

import java.net.\*;

public class InetClient {

public static void main(String[] args) { // runs the Client

String serverName;

if (args.length < 1) serverName = "localhost";

else serverName = args[0];

System.out.println("Clark Elliot's InetClient, 1.8.\n");

System.out.println("Using server: " + serverName + ", Port: 1565");

BufferedReader in = new BufferedReader(new InputStreamReader(System.in)); // creates inputstream for the user

try {

String name;

do {

System.out.print("Enter a hostname or an IP address, (quit) to end: ");

System.out.flush(); //writes the data from the input stream from the user

name = in.readLine();

if (name.indexOf("quit") < 0) // if "quit" isn't inputted, then send a getRemoteAddress signal to the server

getRemoteAddress (name, serverName);

} while (name.indexOf("quit") < 0); //if "quit" is anywhere within the input, close the client.

//This can be an issue with certain websites, such as the example given by Prof Elliot "quitsmoking.com"

System.out.println("Cancelled by user request.");

} catch (IOException x) {x.printStackTrace();}

}

static String toText (byte ip[]) { // Turns an IP address into a String

StringBuffer result = new StringBuffer();

for (int i = 0; i < ip.length; i++) {

if (i > 0) result.append(".");

result.append (0xff & ip[i]);

} return result.toString();

}

static void getRemoteAddress(String name, String serverName) { //contact's the server with the client's request (sends the name the user wrote in the client)

Socket sock;

BufferedReader fromServer;

PrintStream toServer;

String textFromServer;

try {

sock = new Socket(serverName, 1565); // connects to the server

fromServer = new BufferedReader(new InputStreamReader(sock.getInputStream())); //stream recieves input from the server

toServer = new PrintStream(sock.getOutputStream()); //stream sends output to the server

toServer.println(name); toServer.flush(); // sends the data and makes sure of it

for (int i = 1; i <= 3; i++) { // waits for and recieves the input from the server and prints the data from the server

textFromServer = fromServer.readLine();

if (textFromServer != null) System.out.println(textFromServer);

}

sock.close(); // closes the connection to the server

} catch (IOException x) { // catches an IO exception and prints it

System.out.println("Socket error.");

x.printStackTrace();

}

}

}