

```
C:\Users\mirza\Desktop\java1>java KnockKnockClient localhost 3000
cls
Exception in thread "main" java.lang.NumberFormatException: For
input string: "3000cls"
    at java.base/java.lang.NumberFormatException.forInputStr
ing(NumberFormatException.java:67)
    at java.base/java.lang.Integer.parseInt(Integer.java:588
)
    at java.base/java.lang.Integer.parseInt(Integer.java:685
)
    at KnockKnockClient.main(KnockKnockClient.java:13)

C:\Users\mirza\Desktop\java1>java KnockKnockServer 3000

C:\Users\mirza\Desktop\java1>
```

Question: 1
Compiled and executed KnockKnockServer,
KnockKnockClient, and KnockKnockProtocol
using Java Socket and ServerSocket classes.

TCP communication

```
Command Prompt

C:\Users\mirza\Desktop\java1>cd C:\Users\mirza\Desktop\java1

C:\Users\mirza\Desktop\java1>java KnockKnockClient localhost 3000
0
Server: Knock! Knock!
Who's there?
Client: Who's there?
Server: Turnip
Turnip who?
Client: Turnip who?
Server: Turnip the heat, it's cold in here! Want another? (y/n)
y
Client: y
Server: Knock! Knock!
Who's there?
Client: Who's there?
Server: Little Old Lady
Little Old Lady who?
Client: Little Old Lady who?
Server: I didn't know you could yodel! Want another? (y/n)
n
Client: n
Server: Bye.

C:\Users\mirza\Desktop\java1>
```

```
Command Prompt - java Quc × + ▾
Microsoft Windows [Version 10.0.22631.4890]
(c) Microsoft Corporation. All rights reserved.

C:\Users\mirza>cd C:\Users\mirza\Desktop\java

C:\Users\mirza\Desktop\java>javac QuoteServer.java QuoteServerTh
read.java QuoteClient.java

C:\Users\mirza\Desktop\java>java QuoteServer

C:\Users\mirza\Desktop\java>javac QuoteServer.java QuoteServerTh
read.java QuoteClient.java

C:\Users\mirza\Desktop\java>javac QuoteServer.java QuoteServerTh
read.java QuoteClient.java

C:\Users\mirza\Desktop\java>java QuoteServer
QuoteServer: Starting UDP server on port 4445...
QuoteServerThread is listening on UDP port 4445...
Received request from client: /127.0.0.1
Sent quote: Life is wonderful. Without it we'd all be dead.
Received request from client: /127.0.0.1
Sent quote: Daddy, why doesn't this magnet pick up this floppy d
isk?
Received request from client: /127.0.0.1
Sent quote: Give me ambiguity or give me something else.
|
```

```
Command Prompt × + ▾
Microsoft Windows [Version 10.0.22631.4890]
(c) Microsoft Corporation. All rights reserved.

C:\Users\mirza>cd C:\Users\mirza\Desktop\java>j

C:\Users\mirza\Desktop\java>java QuoteClient
Usage: java QuoteClient <hostname>

C:\Users\mirza\Desktop\java>java QuoteClient localhost 4445
Usage: java QuoteClient <hostname>

C:\Users\mirza\Desktop\java>java QuoteClient localhost
Request sent to 127.0.0.1
Quote of the Moment: Life is wonderful. Without it we'd all be d
ead.

C:\Users\mirza\Desktop\java>java QuoteClient localhost
Request sent to 127.0.0.1
Quote of the Moment: Daddy, why doesn't this magnet pick up this
floppy disk?

C:\Users\mirza\Desktop\java>java QuoteClient localhost
Request sent to 127.0.0.1
Quote of the Moment: Give me ambiguity or give me something else
.

C:\Users\mirza\Desktop\java>| Question 2: Created a UDP communication
```

```
(mirza@kali)-[~/Downloads]
$ javac KnockKnockServer.java KnockKnockClient.java KnockKnockProtocol.java
```

Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -Dswing.aatext=true

```
(mirza@kali)-[~/Downloads]
$ java KnockKnockServer 3000
```

Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -Dswing.aatext=true

✓ KnockKnockServer is starting on port 3000 ...

☞ Waiting for a client to connect ...

☞ Client connected from /127.0.0.1

☞ Conversation ended.

```
(mirza@kali)-[~/Downloads]
$ java KnockKnockServer 3000
```

Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -Dswing.aatext=true

✓ KnockKnockServer is starting on port 3000 ...

☞ Waiting for a client to connect ...

☞ Client connected from /127.0.0.1

☞ Conversation ended.

```
(mirza@kali)-[~/Downloads]
$
```

Question 1: In Kali OS

```
(mirza@kali)-[~/Downloads]
$ cd ~/Downloads
```

```
java KnockKnockClient localhost 3000
```

Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -Dswing.aatext=true

Server: Knock! Knock!

Who's there?

Client: Who's there?

Server: Turnip

Turnip who?

Client: Turnip who?

Server: Turnip the heat, it's cold in here! Want another? (y/n)

n

Client: n

Server: Bye.

```
(mirza@kali)-[~/Downloads]
$ cd ~/Downloads
```

```
java KnockKnockClient localhost 3000
```

Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -Dswing.aatext=true

Server: Knock! Knock!

Who's there?

Client: Who's there?

Server: Turnip

Turnip who?

Client: Turnip who?

Server: Turnip the heat, it's cold in here! Want another? (y/n)

y

Client: y

Server: Knock! Knock!

^v

Client:

Server: You're supposed to say "Who's there?"! Try again. Knock! Knock!

ho's there?

Client: ho's there?

Server: You're supposed to say "Who's there?"! Try again. Knock! Knock!

Who's there?

Client: Who's there?

Server: Little Old Lady

Little Old Lady who?

Client: Little Old Lady who?

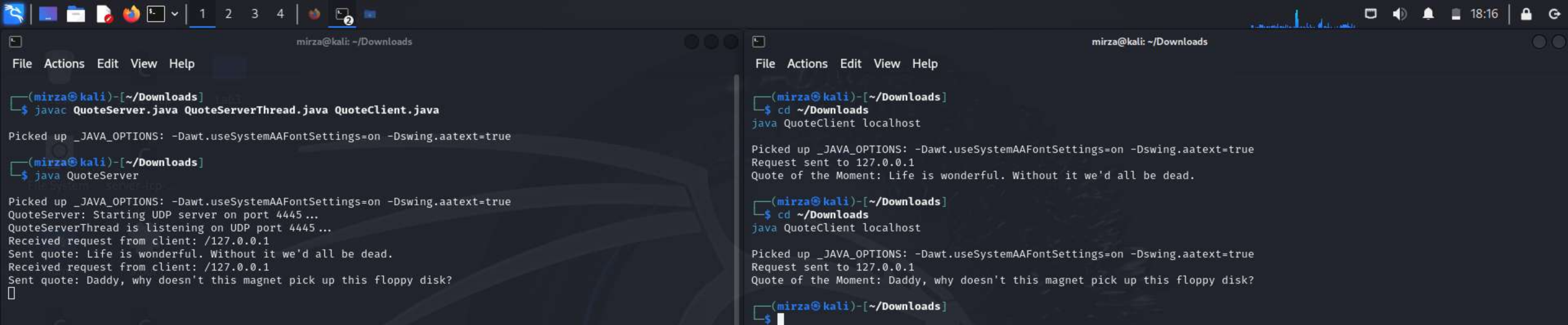
Server: I didn't know you could yodel! Want another? (y/n)

n

Client: n

Server: Bye.

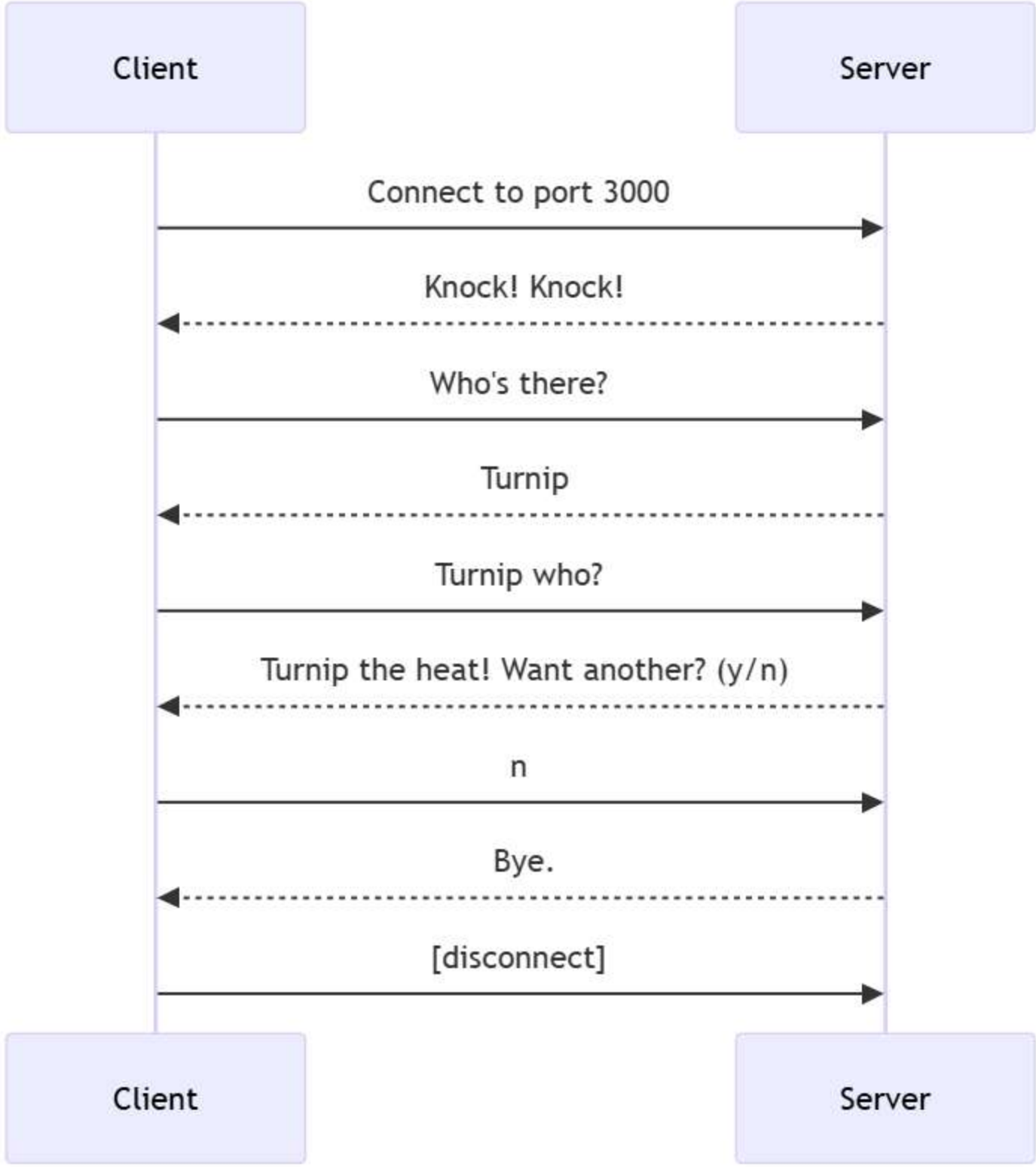
```
(mirza@kali)-[~/Downloads]
$
```

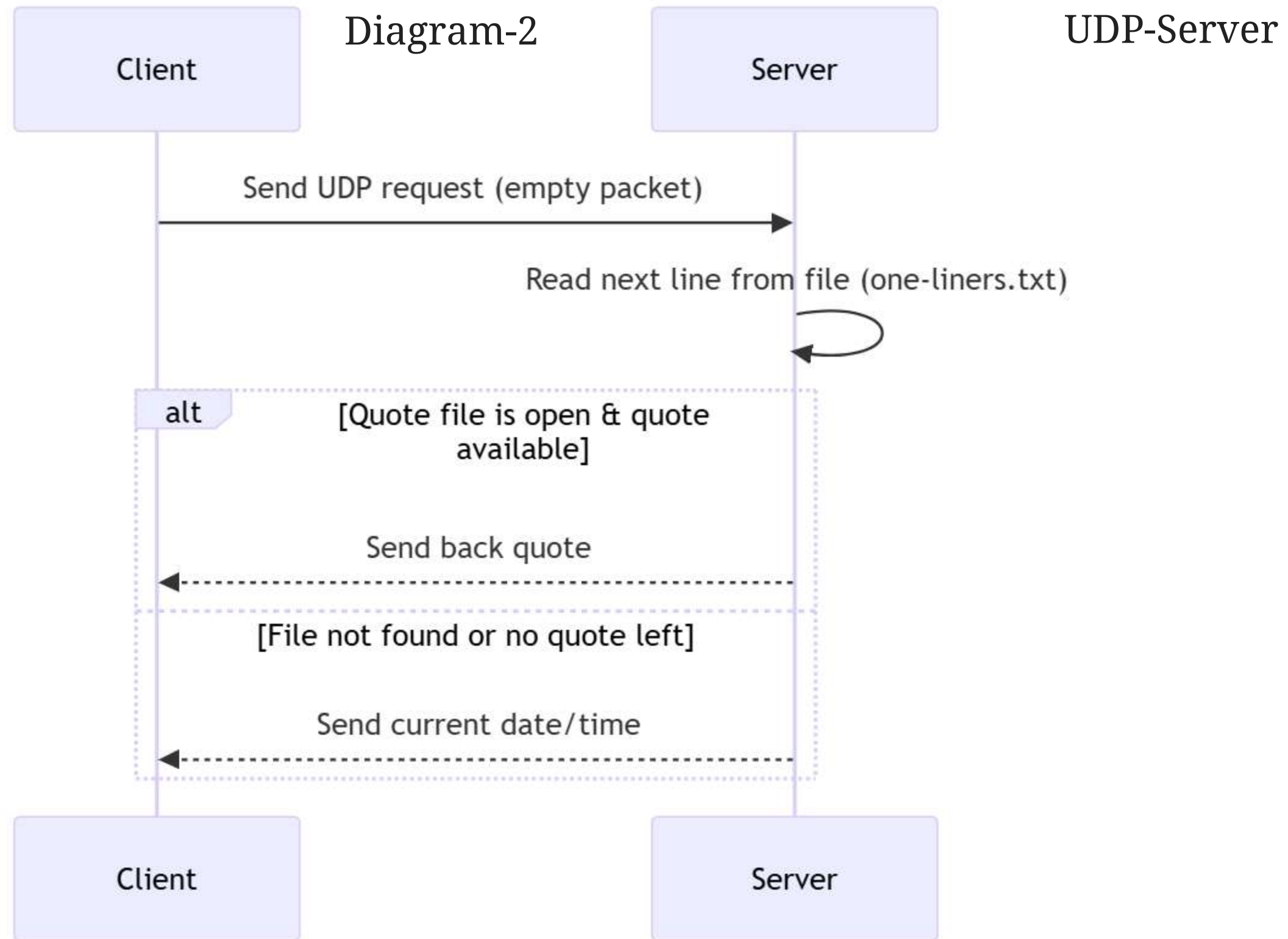



Question 2: In Kali OS

TCP Communication
(KnockKnockServer)

Diagram 1





Comparison: Socket (TCP) vs DatagramSocket (UDP)

Feature	Socket (TCP)	DatagramSocket (UDP)
Connection Type	Connection-oriented	Connectionless
Reliability	Reliable	Unreliable
Ordering	Preserves order	No order guarantee
Speed	Slower	Faster
Overhead	Higher	Lower
Use Cases	Chat, HTTP, File Transfer	Games, Quotes, Streaming
Java Classes	Socket, ServerSocket	DatagramSocket, DatagramPacket

Comparison: Java in Windows vs Kali Linux

Feature	Windows	Kali Linux
File Paths	C:\Users\Name\Folder	/home/user/folder
Java Installation	EXE installer (Oracle/OpenJDK)	`sudo apt install default-jdk`
Terminal Usage	CMD or PowerShell	Bash terminal
Permissions	Standard user	`sudo` needed for ports <1024
Network Tools	Basic (netstat)	Advanced (tcpdump, Wireshark)
Code Execution	`java`, `javac`	`java`, `javac`

KnockKnockServer.java

```
1 | import java.net.*;
2 | import java.io.*;
3 |
4 | public class KnockKnockServer {
5 |     public static void main(String[] args) {
6 |         if (args.length != 1) {
7 |             System.err.println("Usage: java KnockKnockServer <port number>");
8 |             System.exit(1);
9 |         }
10 |
11 |         int portNumber = Integer.parseInt(args[0]);
12 |
13 |         try (
14 |             ServerSocket serverSocket = new ServerSocket(portNumber);
15 |             Socket clientSocket = serverSocket.accept();
16 |             PrintWriter out = new PrintWriter(clientSocket.getOutputStream(),
true);
17 |                 BufferedReader in = new BufferedReader(new
InputStreamReader(clientSocket.getInputStream()))
18 |         ) {
19 |             String inputLine, outputLine;
20 |
21 |             KnockKnockProtocol kkp = new KnockKnockProtocol();
22 |             outputLine = kkp.processInput(null);
23 |             out.println(outputLine);
24 |
25 |             while ((inputLine = in.readLine()) != null) {
26 |                 outputLine = kkp.processInput(inputLine);
27 |                 out.println(outputLine);
28 |                 if ("Bye.".equals(outputLine)) {
29 |                     break;
30 |                 }
31 |             }
32 |
33 |         } catch (IOException e) {
34 |             System.err.println("Error on port " + portNumber + ": " +
e.getMessage());
35 |         }
36 |     }
37 | }
```

KnockKnockProtocol.java

```
1 | public class KnockKnockProtocol {
2 |     private static final int WAITING = 0;
3 |     private static final int SENTKNOCKKNOCK = 1;
4 |     private static final int SENTCLUE = 2;
5 |     private static final int ANOTHER = 3;
6 |
7 |     private static final int NUMJOKES = 5;
8 |
9 |     private int state = WAITING;
10 |    private int currentJoke = 0;
11 |
12 |    private String[] clues = { "Turnip", "Little Old Lady", "Atch", "Who", "Who"
};
13 |    private String[] answers = {
14 |        "Turnip the heat, it's cold in here!",
15 |        "I didn't know you could yodel!",
16 |        "Bless you!",
17 |        "Is there an owl in here?",
18 |        "Is there an echo in here?"
19 |    };
20 |
21 |    public String processInput(String theInput) {
22 |        String theOutput = null;
23 |
24 |        if (state == WAITING) {
25 |            theOutput = "Knock! Knock!";
26 |            state = SENTKNOCKKNOCK;
27 |        } else if (state == SENTKNOCKKNOCK) {
28 |            if (theInput.equalsIgnoreCase("Who's there?")) {
29 |                theOutput = clues[currentJoke];
30 |                state = SENTCLUE;
31 |            } else {
32 |                theOutput = "You're supposed to say \"Who's there?\"! Try again.
Knock! Knock!";
33 |            }
34 |        } else if (state == SENTCLUE) {
35 |            if (theInput.equalsIgnoreCase(clues[currentJoke] + " who?")) {
36 |                theOutput = answers[currentJoke] + " Want another? (y/n)";
37 |                state = ANOTHER;
38 |            } else {
39 |                theOutput = "You're supposed to say \"" + clues[currentJoke] + "
who?\" Try again. Knock! Knock!";
40 |                state = SENTKNOCKKNOCK;
41 |            }
42 |        } else if (state == ANOTHER) {
43 |            if (theInput.equalsIgnoreCase("y")) {
44 |                theOutput = "Knock! Knock!";
45 |                currentJoke = (currentJoke + 1) % NUMJOKES;
46 |                state = SENTKNOCKKNOCK;
47 |            } else {
48 |                theOutput = "Bye.";
```

KnockKnockProtocol.java

```
49 |         state = WAITING;
50 |     }
51 | }
52 |
53 |     return theOutput;
54 | }
55 | }
```

KnockKnockClient.java

```
1 | import java.io.*;
2 | import java.net.*;
3 |
4 | public class KnockKnockClient {
5 |     public static void main(String[] args) throws IOException {
6 |
7 |         if (args.length != 2) {
8 |             System.err.println("Usage: java KnockKnockClient <host name> <port
number>");
9 |             System.exit(1);
10 |        }
11 |
12 |        String hostName = args[0];
13 |        int portNumber = Integer.parseInt(args[1]);
14 |
15 |        try (
16 |            Socket kkSocket = new Socket(hostName, portNumber);
17 |            PrintWriter out = new PrintWriter(kkSocket.getOutputStream(), true);
18 |            BufferedReader in = new BufferedReader(
19 |                new InputStreamReader(kkSocket.getInputStream()));
20 |        ) {
21 |            BufferedReader stdIn = new BufferedReader(new
InputStreamReader(System.in));
22 |            String fromServer;
23 |            String fromUser;
24 |
25 |            while ((fromServer = in.readLine()) != null) {
26 |                System.out.println("Server: " + fromServer);
27 |                if (fromServer.equals("Bye."))
28 |                    break;
29 |
30 |                fromUser = stdIn.readLine();
31 |                if (fromUser != null) {
32 |                    System.out.println("Client: " + fromUser);
33 |                    out.println(fromUser);
34 |                }
35 |            }
36 |        } catch (UnknownHostException e) {
37 |            System.err.println("Don't know about host " + hostName);
38 |            System.exit(1);
39 |        } catch (IOException e) {
40 |            System.err.println("Couldn't get I/O for the connection to " +
hostName);
41 |            System.exit(1);
42 |        }
43 |    }
44 | }
```

QuoteServer.java

```
1 | import java.io.*;
2 |
3 | public class QuoteServer {
4 |     public static void main(String[] args) throws IOException {
5 |         System.out.println("QuoteServer: Starting UDP server on port 4445...");
6 |         new QuoteServerThread().start();
7 |     }
8 | }
```


QuoteServerThread.java

```
1 | import java.io.*;
2 | import java.net.*;
3 | import java.util.*;
4 |
5 | public class QuoteServerThread extends Thread {
6 |
7 |     protected DatagramSocket socket = null;
8 |     protected BufferedReader in = null;
9 |     protected boolean moreQuotes = true;
10 |
11 |     public QuoteServerThread() throws IOException {
12 |         this("QuoteServerThread");
13 |     }
14 |
15 |     public QuoteServerThread(String name) throws IOException {
16 |         super(name);
17 |         socket = new DatagramSocket(4445);
18 |
19 |         System.out.println("QuoteServerThread is listening on UDP port 4445...");
20 |
21 |         try {
22 |             in = new BufferedReader(new FileReader("one-liners.txt"));
23 |         } catch (FileNotFoundException e) {
24 |             System.err.println("Could not open quote file. Serving time
instead.");
25 |         }
26 |     }
27 |
28 |     public void run() {
29 |         while (moreQuotes) {
30 |             try {
31 |                 byte[] buf = new byte[256];
32 |
33 |                 // receive request
34 |                 DatagramPacket packet = new DatagramPacket(buf, buf.length);
35 |                 socket.receive(packet);
36 |                 System.out.println("Received request from client: " +
packet.getAddress());
37 |
38 |                 // prepare response
39 |                 String dString = (in == null) ? new Date().toString() :
getNextQuote();
40 |                 buf = dString.getBytes();
41 |
42 |                 // send response
43 |                 InetAddress address = packet.getAddress();
44 |                 int port = packet.getPort();
45 |                 packet = new DatagramPacket(buf, buf.length, address, port);
46 |                 socket.send(packet);
47 |                 System.out.println("Sent quote: " + dString);
48 |             } catch (IOException e) {
```

QuoteServerThread.java

```
49 |         e.printStackTrace();
50 |         moreQuotes = false;
51 |     }
52 | }
53 | socket.close();
54 | }
55 |
56 | protected String getNextQuote() {
57 |     String returnValue;
58 |     try {
59 |         returnValue = in.readLine();
60 |         if (returnValue == null) {
61 |             in.close();
62 |             moreQuotes = false;
63 |             returnValue = "No more quotes. Goodbye.";
64 |         }
65 |     } catch (IOException e) {
66 |         returnValue = "IOException occurred in server.";
67 |     }
68 |     return returnValue;
69 | }
70 | }
```

QuoteClient.java

```
1 | import java.io.*;
2 | import java.net.*;
3 |
4 | public class QuoteClient {
5 |     public static void main(String[] args) throws IOException {
6 |         if (args.length != 1) {
7 |             System.out.println("Usage: java QuoteClient <hostname>");
8 |             return;
9 |         }
10 |
11 |         // get a datagram socket
12 |         DatagramSocket socket = new DatagramSocket();
13 |
14 |         // send request
15 |         byte[] buf = new byte[256];
16 |         InetAddress address = InetAddress.getByName(args[0]);
17 |         DatagramPacket packet = new DatagramPacket(buf, buf.length, address,
4445);
18 |         socket.send(packet);
19 |         System.out.println("Request sent to " + address.getHostAddress());
20 |
21 |         // get response
22 |         packet = new DatagramPacket(buf, buf.length);
23 |         socket.receive(packet);
24 |
25 |         // display response
26 |         String received = new String(packet.getData(), 0, packet.getLength());
27 |         System.out.println("Quote of the Moment: " + received);
28 |
29 |         socket.close();
30 |     }
31 | }
```

one-liners.txt

1 | Life is wonderful. Without it we'd all be dead.
2 | Daddy, why doesn't this magnet pick up this floppy disk?
3 | Give me ambiguity or give me something else.
4 | I.R.S.: We've got what it takes to take what you've got!
5 | We are born naked, wet and hungry. Then things get worse.
6 | Make it idiot proof and someone will make a better idiot.
7 | He who laughs last thinks slowest!
8 | Always remember you're unique, just like everyone else.
9 | "More hay, Trigger?" "No thanks, Roy, I'm stuffed!"
10 | A flashlight is a case for holding dead batteries.
11 | Lottery: A tax on people who are bad at math.
12 | Error, no keyboard - press F1 to continue.
13 | There's too much blood in my caffeine system.
14 | Artificial Intelligence usually beats real stupidity.
15 | Hard work has a future payoff. Laziness pays off now.
16 | "Very funny, Scotty. Now beam down my clothes."
17 | Puritanism: The haunting fear that someone, somewhere may be happy.
18 | Consciousness: that annoying time between naps.
19 | Don't take life too seriously, you won't get out alive.
20 | I don't suffer from insanity. I enjoy every minute of it.
21 | Better to understand a little than to misunderstand a lot.
22 | The gene pool could use a little chlorine.
23 | When there's a will, I want to be in it.
24 | Okay, who put a "stop payment" on my reality check?
25 | We have enough youth, how about a fountain of SMART?
26 | Programming is an art form that fights back.
27 | "Daddy, what does FORMATTING DRIVE C mean?"
28 | All wiyht. Rho sritched mg kegtops awound?
29 | My mail reader can beat up your mail reader.
30 | Never forget: $2 + 2 = 5$ for extremely large values of 2.
31 | Nobody has ever, ever, EVER learned all of WordPerfect.
32 | To define recursion, we must first define recursion.
33 | Good programming is 99% sweat and 1% coffee.
34 | Home is where you hang your @
35 | The E-mail of the species is more deadly than the mail.
36 | A journey of a thousand sites begins with a single click.
37 | You can't teach a new mouse old clicks.
38 | Great groups from little icons grow.
39 | Speak softly and carry a cellular phone.
40 | C:\ is the root of all directories.
41 | Don't put all your hypes in one home page.
42 | Pentium wise; pen and paper foolish.
43 | The modem is the message.
44 | Too many clicks spoil the browse.
45 | The geek shall inherit the earth.
46 | A chat has nine lives.
47 | Don't byte off more than you can view.
48 | Fax is stranger than fiction.
49 | What boots up must come down.
50 | Windows will never cease. (ed. oh sure...)
51 | In Gates we trust. (ed. yeah right....)

one-liners.txt

52 | Virtual reality is its own reward.
53 | Modulation in all things.
54 | A user and his leisure time are soon parted.
55 | There's no place like <http://www.home.com>
56 | Know what to expect before you connect.
57 | Oh, what a tangled website we weave when first we practice.
58 | Speed thrills.
59 | Give a man a fish and you feed him for a day; teach him to use the Net and he won't bother you for weeks.