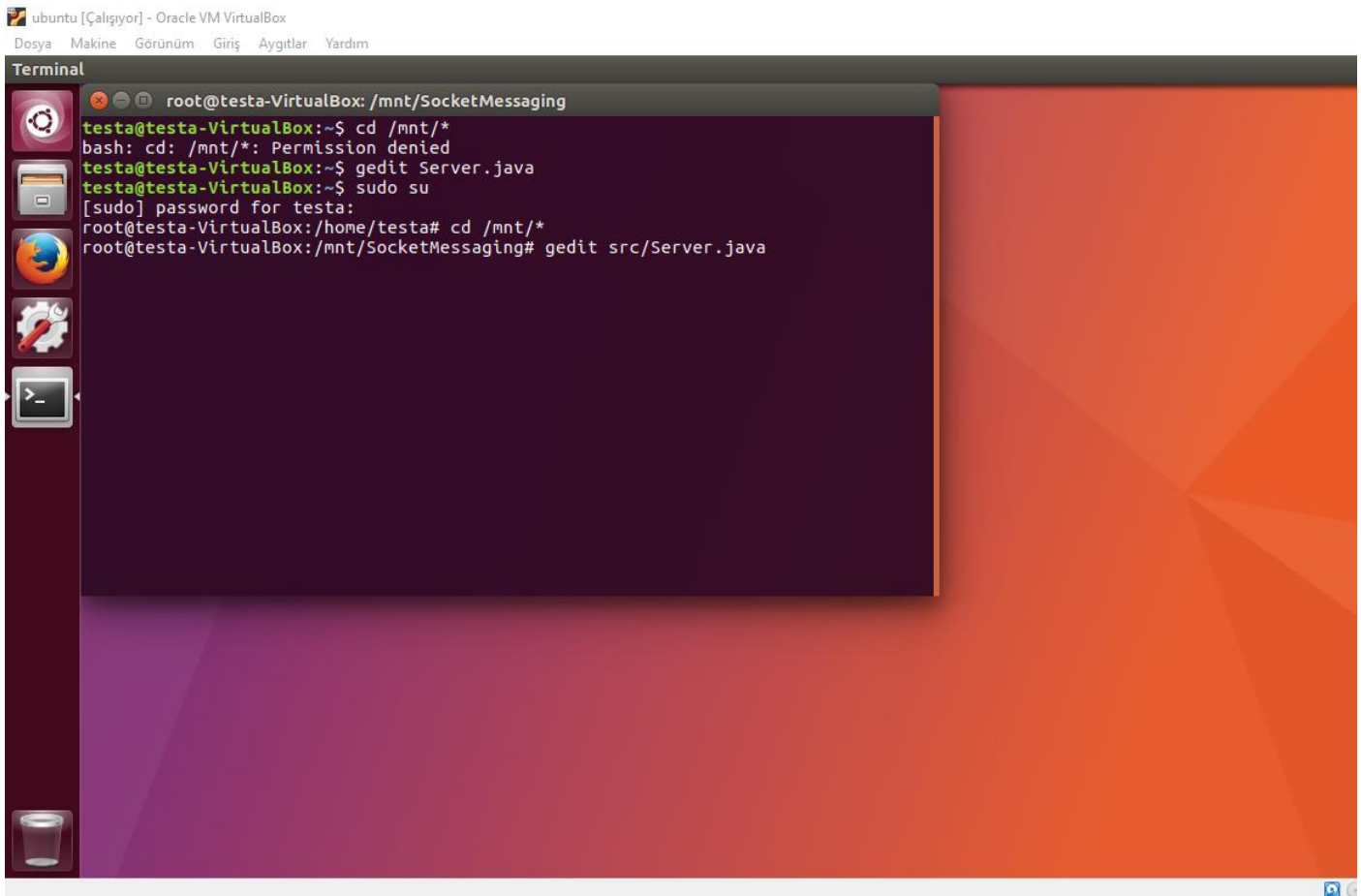


# CLIENT SERVER MESSENGER

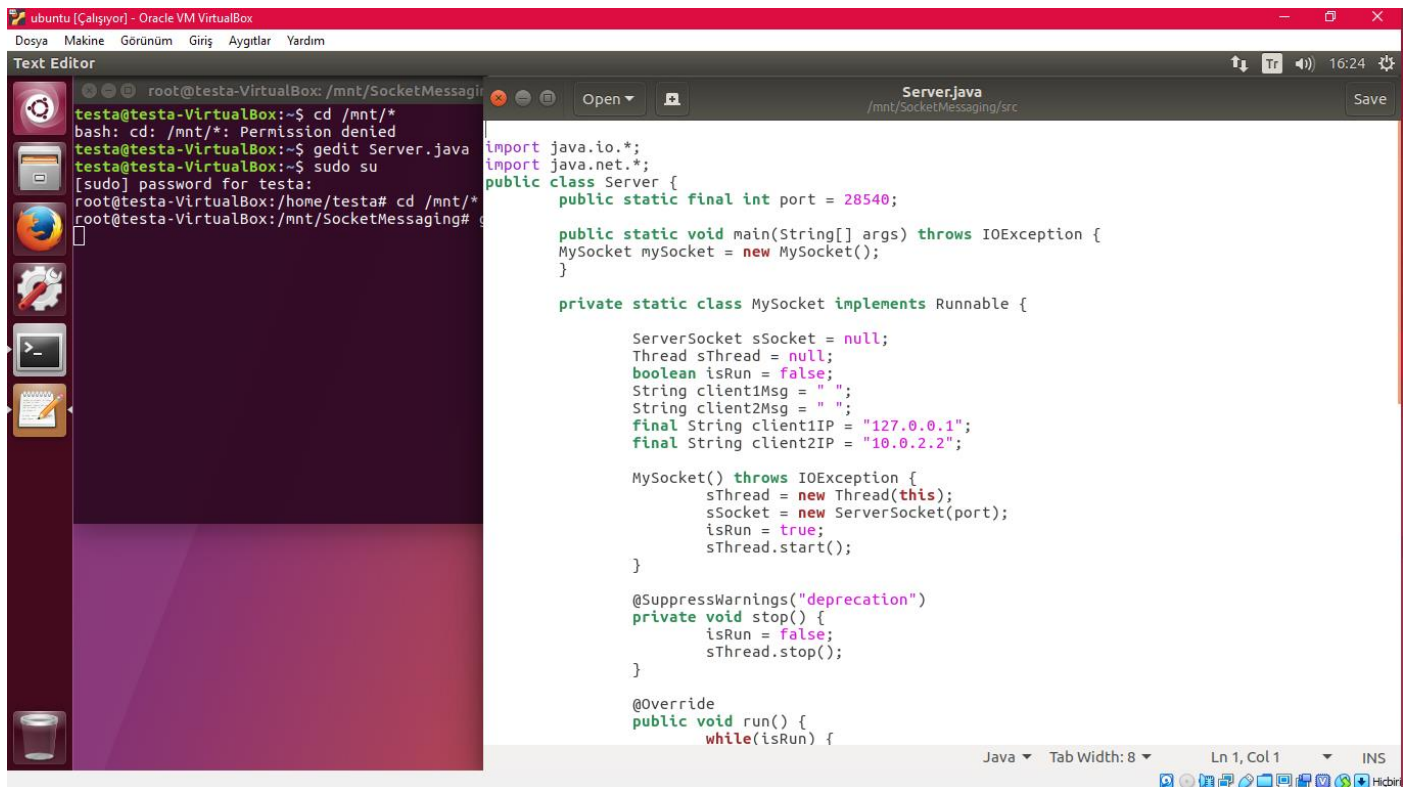
Firstly, We should be a super user to connect 'mnt' file which has java classes of socket messaging on our virtual machine (Ubuntu 17.04). For this reason, we used command that sudo su and enter our password.

Now, we are a super user and connect to mnt file with command that cd /mnt/\*.



```
ubuntu [Çalışıyor] - Oracle VM VirtualBox
Dosya Makine Görünüm Giriş Aygıtlar Yardım
Terminal
root@testa-VirtualBox: /mnt/SocketMessaging
testa@testa-VirtualBox:~$ cd /mnt/*
bash: cd: /mnt/*: Permission denied
testa@testa-VirtualBox:~$ gedit Server.java
testa@testa-VirtualBox:~$ sudo su
[sudo] password for testa:
root@testa-VirtualBox:/home/testa# cd /mnt/*
root@testa-VirtualBox:/mnt/SocketMessaging# gedit src/Server.java
```

Next, we wanted to see codes of the server class. For this reason, we used the command that 'gedit src/Server.java' to open server class.



```
root@testa-VirtualBox: /mnt/SocketMessaging
testa@testa-VirtualBox:~$ cd /mnt/*
bash: cd: /mnt/*: Permission denied
testa@testa-VirtualBox:~$ gedit Server.java
testa@testa-VirtualBox:~$ sudo su
[sudo] password for testa:
root@testa-VirtualBox:/home/testa# cd /mnt/*
root@testa-VirtualBox:/mnt/SocketMessaging# gedit src/Server.java
```

```
Server.java
/mnt/SocketMessaging/src

import java.io.*;
import java.net.*;

public class Server {
    public static final int port = 28540;

    public static void main(String[] args) throws IOException {
        MySocket mySocket = new MySocket();
    }

    private static class MySocket implements Runnable {

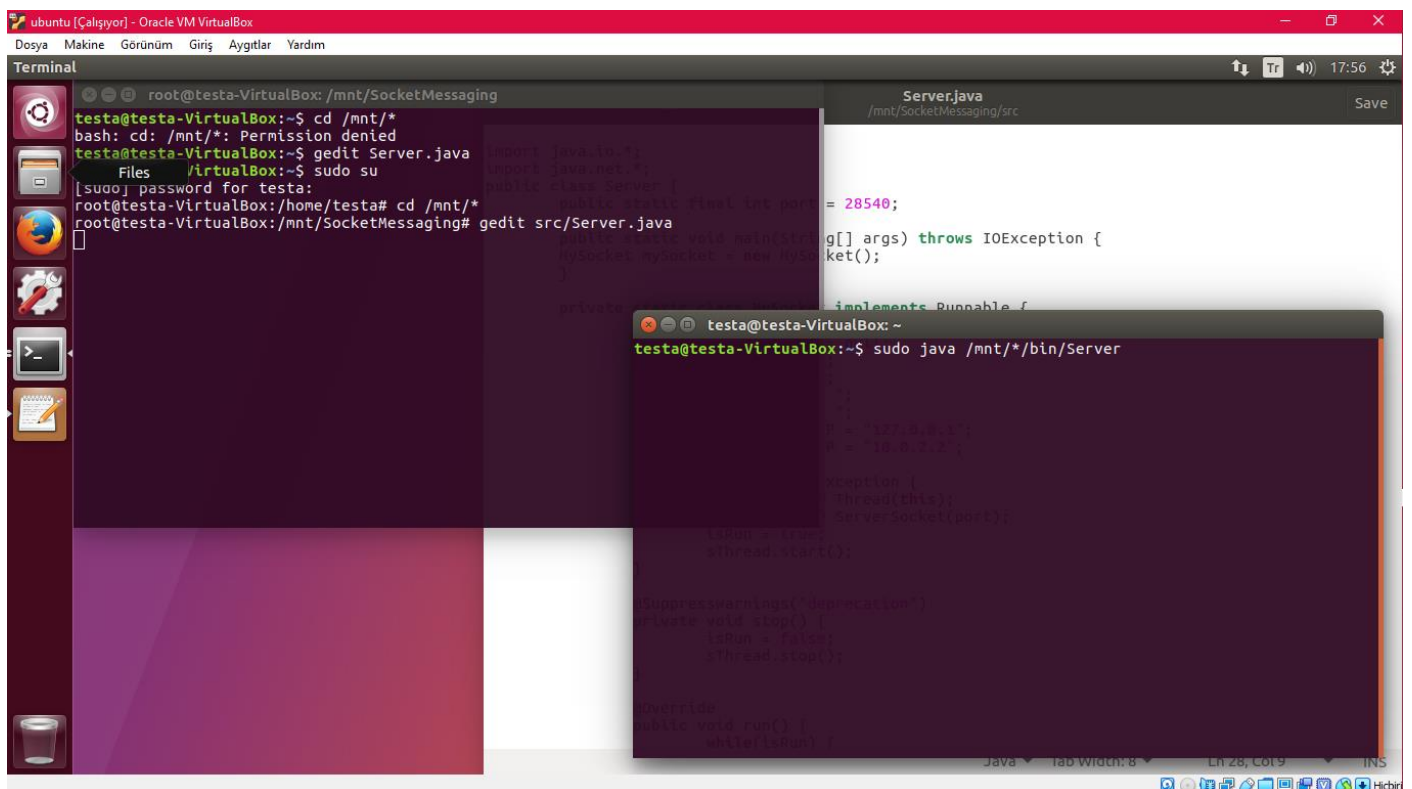
        ServerSocket sSocket = null;
        Thread sThread = null;
        boolean isRun = false;
        String client1Msg = " ";
        String client2Msg = " ";
        final String client1IP = "127.0.0.1";
        final String client2IP = "10.0.2.2";

        MySocket() throws IOException {
            sThread = new Thread(this);
            sSocket = new ServerSocket(port);
            isRun = true;
            sThread.start();
        }

        @SuppressWarnings("deprecation")
        private void stop() {
            isRun = false;
            sThread.stop();
        }

        @Override
        public void run() {
            while(isRun) {
```

Secondly, we opened new terminal and running server with using command that Java /mnt/\*/bin/Server



```
root@testa-VirtualBox: /mnt/SocketMessaging
testa@testa-VirtualBox:~$ cd /mnt/*
bash: cd: /mnt/*: Permission denied
testa@testa-VirtualBox:~$ gedit Server.java
Files /VirtualBox:~$ sudo su
[sudo] password for testa:
root@testa-VirtualBox:/home/testa# cd /mnt/*
root@testa-VirtualBox:/mnt/SocketMessaging# gedit src/Server.java
```

```
Server.java
/mnt/SocketMessaging/src

import java.io.*;
import java.net.*;

public class Server {
    public static final int port = 28540;

    public static void main(String[] args) throws IOException {
        MySocket mySocket = new MySocket();
    }

    private static class MySocket implements Runnable {

        ServerSocket sSocket = null;
        Thread sThread = null;
        boolean isRun = false;
        String client1Msg = " ";
        String client2Msg = " ";
        final String client1IP = "127.0.0.1";
        final String client2IP = "10.0.2.2";

        MySocket() throws IOException {
            sThread = new Thread(this);
            sSocket = new ServerSocket(port);
            isRun = true;
            sThread.start();
        }

        @SuppressWarnings("deprecation")
        private void stop() {
            isRun = false;
            sThread.stop();
        }

        @Override
        public void run() {
            while(isRun) {
```

```
testa@testa-VirtualBox: ~
testa@testa-VirtualBox:~$ sudo java /mnt/*/bin/Server
```

Then, Running the Client 2 at Eclipse. At the same time, Client 1 connected to the ubuntu server on virtual machine.

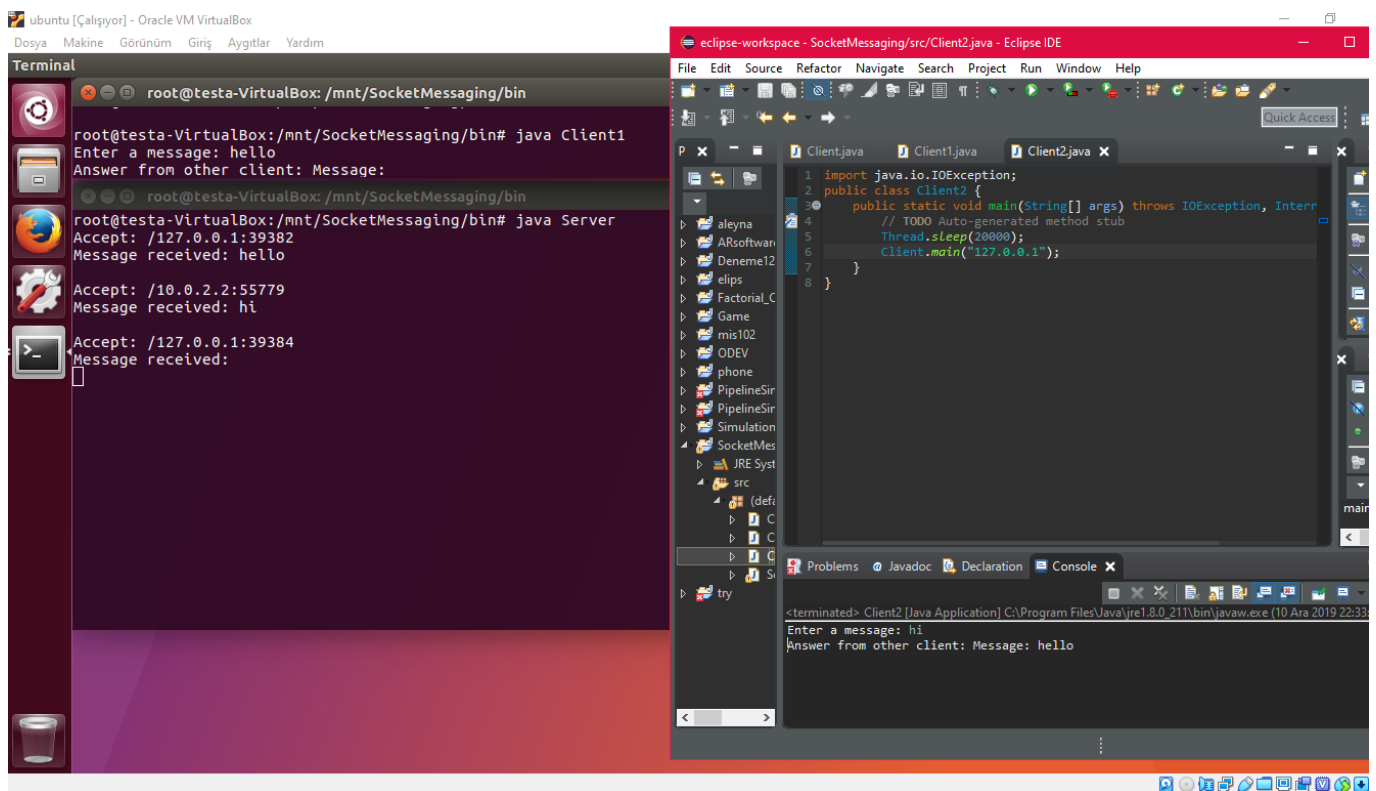
Client 1 send a message that Hello.

The message received.

Client 2 connected to the ubuntu server on virtual machine.

Client 2 received the message from Client 1.

Sending a message from Client 2 to Client 1.This message is Hi.



The screenshot displays a virtual machine environment with two main windows: a terminal and an Eclipse IDE.

**Terminal Window:** The terminal shows the execution of a Java server and two clients. The commands and their outputs are as follows:

```
root@testa-VirtualBox: /mnt/SocketMessaging/bin# java Client1
Enter a message: hello
Answer from other client: Message:

root@testa-VirtualBox: /mnt/SocketMessaging/bin# java Server
Message received: hello
Accept: /10.0.2.2:55779
Message received: hi
Accept: /127.0.0.1:39384
Message received:
```

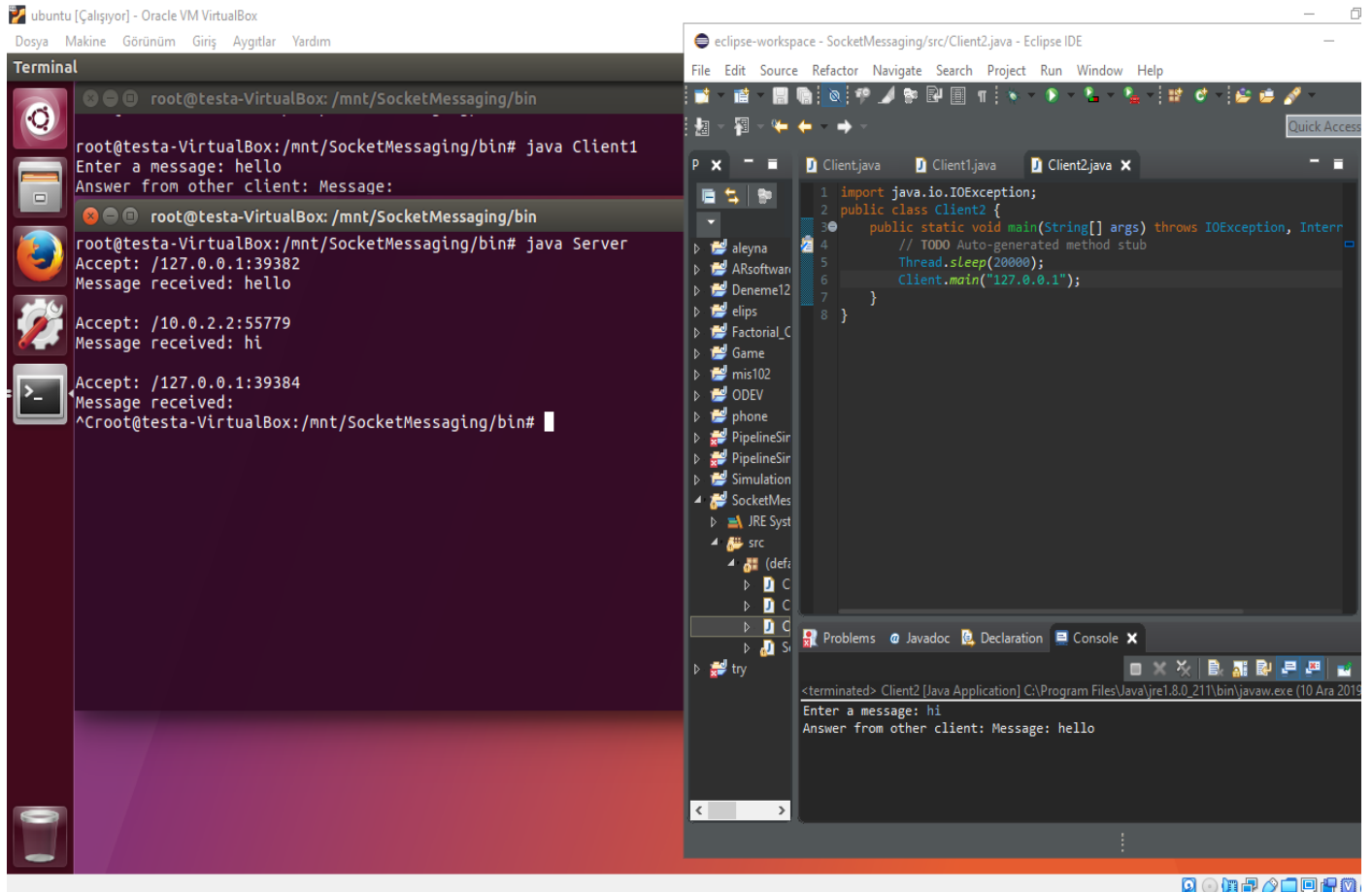
**Eclipse IDE Window:** The Eclipse IDE shows the source code for `Client2.java`. The code is as follows:

```
1 import java.io.IOException;
2 public class Client2 {
3     public static void main(String[] args) throws IOException, Interr
4         // TODO Auto-generated method stub
5         Thread.sleep(20000);
6         Client.main("127.0.0.1");
7     }
8 }
```

The Eclipse IDE also shows a console window with the following output:

```
<terminated> Client2 [Java Application] C:\Program Files\Java\jre1.8.0_211\bin\javaw.exe (10 Ara 2019 22:33)
Enter a message: hi
Answer from other client: Message: hello
```

Finally, we should close the connection. We wrote command that  
bin# .



The screenshot displays a terminal window on the left and an Eclipse IDE on the right. The terminal window, titled 'ubuntu [Çalışıyor] - Oracle VM VirtualBox', shows the execution of a Java application. The user is in the directory '/mnt/SocketMessaging/bin'. They run 'java Client1', which prompts 'Enter a message: hello' and then 'Answer from other client: Message:'. Next, they run 'java Server', which shows 'Accept: /127.0.0.1:39382', 'Message received: hello', 'Accept: /10.0.2.2:55779', 'Message received: hi', and 'Accept: /127.0.0.1:39384', 'Message received:'. The terminal prompt is '^Croot@testa-VirtualBox:/mnt/SocketMessaging/bin#'. The Eclipse IDE, titled 'eclipse-workspace - SocketMessaging/src/Client2.java - Eclipse IDE', shows the source code for 'Client2.java'. The code imports 'java.io.IOException' and defines a 'Client2' class with a 'main' method. The 'main' method calls 'Client.main("127.0.0.1")'. The IDE's console window at the bottom shows the output of the application: '<terminated> Client2 [Java Application] C:\Program Files\Java\jre1.8.0\_211\bin\javaw.exe (10 Ara 2019)', 'Enter a message: hi', and 'Answer from other client: Message: hello'.

```
root@testa-VirtualBox:/mnt/SocketMessaging/bin# java Client1
Enter a message: hello
Answer from other client: Message:

root@testa-VirtualBox:/mnt/SocketMessaging/bin# java Server
Accept: /127.0.0.1:39382
Message received: hello

Accept: /10.0.2.2:55779
Message received: hi

Accept: /127.0.0.1:39384
Message received:
^Croot@testa-VirtualBox:/mnt/SocketMessaging/bin#
```

```
1 import java.io.IOException;
2 public class Client2 {
3     public static void main(String[] args) throws IOException, Interr
4         // TODO Auto-generated method stub
5         Thread.sleep(20000);
6         Client.main("127.0.0.1");
7     }
8 }
```

```
<terminated> Client2 [Java Application] C:\Program Files\Java\jre1.8.0_211\bin\javaw.exe (10 Ara 2019)
Enter a message: hi
Answer from other client: Message: hello
```