

Network Architecture-1

Project-2

Fall 2016



Submitted by:

Moulika Chadalavada (16234180)

Lakshmi Nikitha Kona (16231555)

Sri Sai Anusha Gandu (16230560)

Table of Contents

1. Introduction	2
2. About GENI and Putty	2
3. Project Description	7
3.1 Question-1	7
3.2 Question-2	9
3.3 Question-3	12
3.4 Question-4	15
4. References	19

1. Introduction

In this project, we are developing a simple Chat application. TCP Client and Server programs are executed for communication between resources using GENI. In Question 1, we create a chat server which accepts a single client connection and displays everything the client types and the communication get terminated from both sides when the client types 'exit'. In Question 2, extending the Question 1, when one client terminates, the server remains open for the other client's other clients that wish to communicate and can handle only one connection at a time. In Question 3, extending the Question 2, the server will be able to handle multiple clients at a time and the server window displays the messages from the clients. In Question 4, extending the Question 3, the server even echoes the messages from a client to all the remaining clients in the network.

2. About GENI and Putty

To continue with the project, we need to first create a GENI account and a slice where we can reserve resources on which to work on.

GENI Account Creation:

GENI (Global Environment for Network Innovations) provides a virtual laboratory for networking and distributed systems research and education. It is well suited for exploring networks at scale, thereby promoting innovations in network science, security, services and applications.

1. Login to portal.geni.net
2. Activate the GENI account with the required credentials.
3. Download the SSH keys (Putty) for authentication process.
4. Using the Putty key generator, generate a private key which will be used to open the Client and Server windows.

SSH Keys

Name	Description	Public Key	Private Key	PuTTY	Edit	Delete
id_geni_ssh_rsa 49:4f:30:da:25:e9:6d:d1:2a:8b:ae:5d:cd:70:20:5f	Generated SSH keypair	Download Public Key	Download Private Key	Download PuTTY Key	Edit	Delete

On Linux and Mac systems and for most Windows SSH clients (not PuTTY), do:

- Download your private key.
- On Windows, just point your SSH client (not PuTTY) to the downloaded private key.
- On Linux and Mac, open a terminal.
 - Store your key under `~/.ssh/` :
 - If the directory does not exist, create it:


```
mkdir ~/.ssh
```
 - Move the key to `~/.ssh/` :


```
mv ~/Downloads/id_geni_ssh_rsa ~/.ssh/
```
 - Change the file permissions:


```
chmod 0600 ~/.ssh/id_geni_ssh_rsa
```
- Your SSH command will be something like:


```
ssh -i ~/.ssh/id_geni_ssh_rsa [username]@[hostname] -p [port]
```

For PuTTY users:

SSH Keys

Slice Creation:

Create a slice from where the resources can be reserved. For this project, New Slice 'NA-Project2' is created, where initially there are no resources which are reserved.

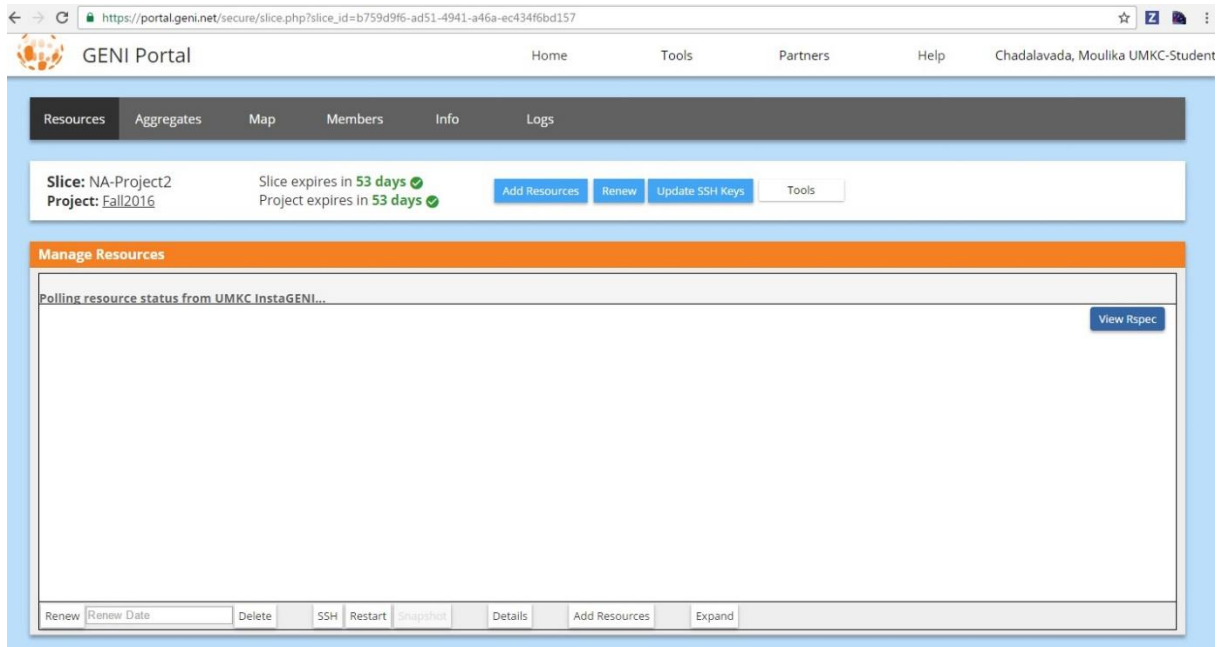
Slices

Filter by: Sort by: ☒ Sort ascending

[+ New slice](#)

NA-Project2	Project-NA-1
Project: Fall2016	Project: Fall2016
Owner: Chadalavada, Moulika UMKC-Stu...	Owner: Chadalavada, Moulika UMKC-Stu...
Slice expires in 53 days <input checked="" type="checkbox"/>	Slice expires in 53 days <input checked="" type="checkbox"/>
No resources for this slice	3 resources, next exp. in 53 days <input checked="" type="checkbox"/>

Slice NA-Project2 Creation

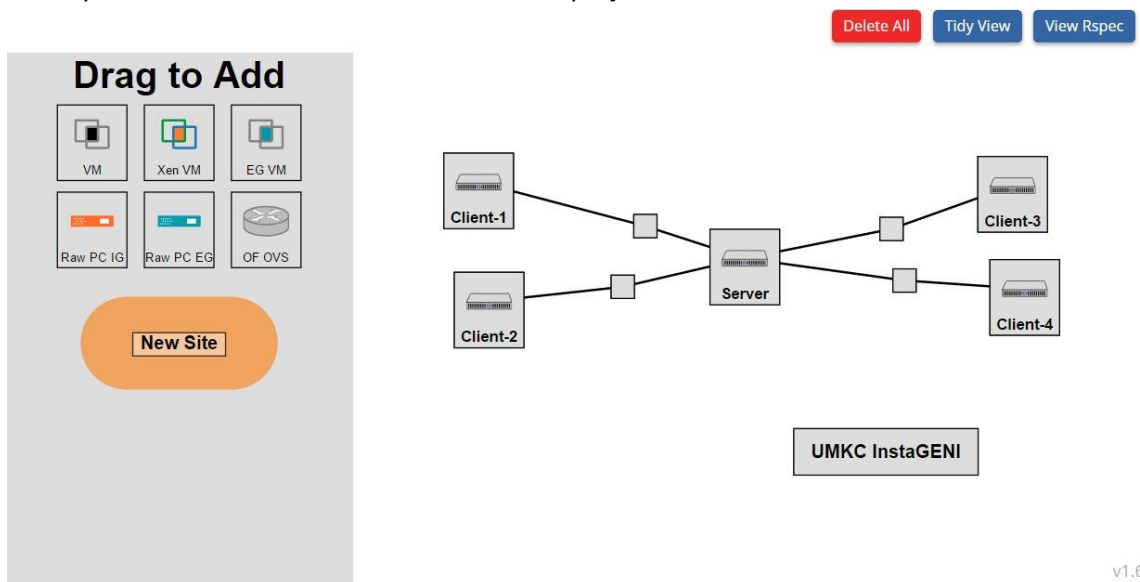


No Resources under Slice

Resource Reservation:

In the Slice under Add Resources, where VM's are added named them as 'n' Clients and Server. Also, established connection (link) between them and provided the IP address and subnet mask along with bandwidth in the link interface.

Initially 4 Client's and 1 Server is added for this project and Reserved these resources.



Adding VM's under Slice

After adding the resources the status is unknown and later the resources are ready as shown below.

The screenshot shows the GENI Portal interface. At the top, there's a navigation bar with links: Home, Tools, Partners, Help, and a user profile 'Chadalavada, Moulika UMKC-Student'. Below this is a secondary navigation bar with links: Resources, Aggregates, Map, Members, Info, and Logs. The main content area has a header for 'Slice: NA-Project2' and 'Project: Fall2016', with expiration dates 'Slice expires in 53 days' and 'Project expires in 53 days', both marked with green checkmarks. Action buttons include 'Add Resources', 'Renew', 'Update SSH Keys', and 'Tools'. The 'Manage Resources' section is highlighted with an orange header. It contains a message 'Polling resource status from UMKC InstaGENI...' and a 'View Rspec' button. The network diagram shows a central 'Server' node connected to four 'Client' nodes (Client-1, Client-2, Client-3, Client-4). The diagram is rendered in a light gray, indicating that the resources are in an 'unknown' state. At the bottom of the diagram area, there's a row of buttons: Renew, Renew Date, Delete, SSH, Restart, Snapshot, Details, Add Resources, and Expand.

Resources when status is unknown

This screenshot shows the same GENI Portal interface as the previous one, but the status of the resources has changed. The message in the 'Manage Resources' section now reads 'Resources on UMKC InstaGENI are ready.' The network diagram shows the same topology (Server connected to Client-1, Client-2, Client-3, and Client-4), but all nodes are now highlighted in green, indicating they are in a 'ready' state. The 'View Rspec' button and the bottom row of buttons remain the same.

When all the resources are ready

GENI Portal Home Tools Partners Help Chadalavada, Moulika UMKC-Stud

Resources on slice: NA-Project2

Queried 1 of 1 aggregates.

[Refresh All Details](#) [Refresh Status Only](#)

Status	Aggregate
READY	UMKC InstaGENI

Aggregate **UMKC InstaGENI's** Resources:

Node #1:

Status	Client ID	Component ID	Expiration	Type	Hostname
READY	Client-1	pc3	2016-12-30T23:59:59.000Z	default-vm	Client-1.NA-Project2.ch-geni-net.instageni.umkc.edu
Login	ssh hhgc77@pc3.instageni.umkc.edu -p 33594				
	ssh choibv@pc3.instageni.umkc.edu -p 33594				
	ssh rzcd46@pc3.instageni.umkc.edu -p 33594				
	ssh mc7d8@pc3.instageni.umkc.edu -p 33594				
Interfaces		MAC	Layer 3		
Interface-0		pc3:1a0	02ef71764fe9		
			IPv4: 10.10.1.1		

Node #2:

Status	Client ID	Component ID	Expiration	Type	Hostname
READY	Client-2	pc3	2016-12-30T23:59:59.000Z	default-vm	Client-2.NA-Project2.ch-geni-net.instageni.umkc.edu
Login	ssh hhgc77@pc3.instageni.umkc.edu -p 33595				
	ssh choibv@pc3.instageni.umkc.edu -p 33595				
	ssh rzcd46@pc3.instageni.umkc.edu -p 33595				
	ssh mc7d8@pc3.instageni.umkc.edu -p 33595				
Interfaces		MAC	Layer 3		
Interface-0		pc3:1a0	02ef71764fe9		
			IPv4: 10.10.1.1		

All Client's and Server resources are Ready

Once the resources are created in Slice and SSH keys are downloaded, in **puttygen** private key is generated using Putty key are is downloaded from GENI. Once Private key is generated in putty, logged in to VM's of Clients and Server with its respective Host Name and Port Number. Also before logging in Private key is loaded under SSH -> Auth.

Name

Server

SSH to

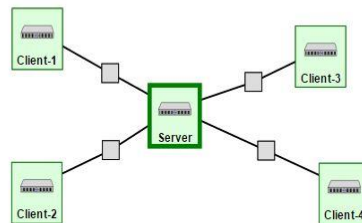
hhgc77@pc3.instageni.umkc.edu:33598,
choibv@pc3.instageni.umkc.edu:33598,
rzcd46@pc3.instageni.umkc.edu:33598,
mc7d8@pc3.instageni.umkc.edu:33598

Node Type

Other...

default-vm

Hardware Type



PuTTY Configuration

Category: Session

Basic options for your PuTTY session

Specify the destination you want to connect to

Host Name (or IP address) Port

mc7d8@pc3.instageni.umkc.edu 33598

Connection type:

☐ Raw ☐ Telnet ☐ Rlogin ☒ SSH ☐ Serial

Load, save or delete a stored session

Saved Sessions

Server2

Default Settings

Client

Client-1

Client-2

Client-3

Server

Server2

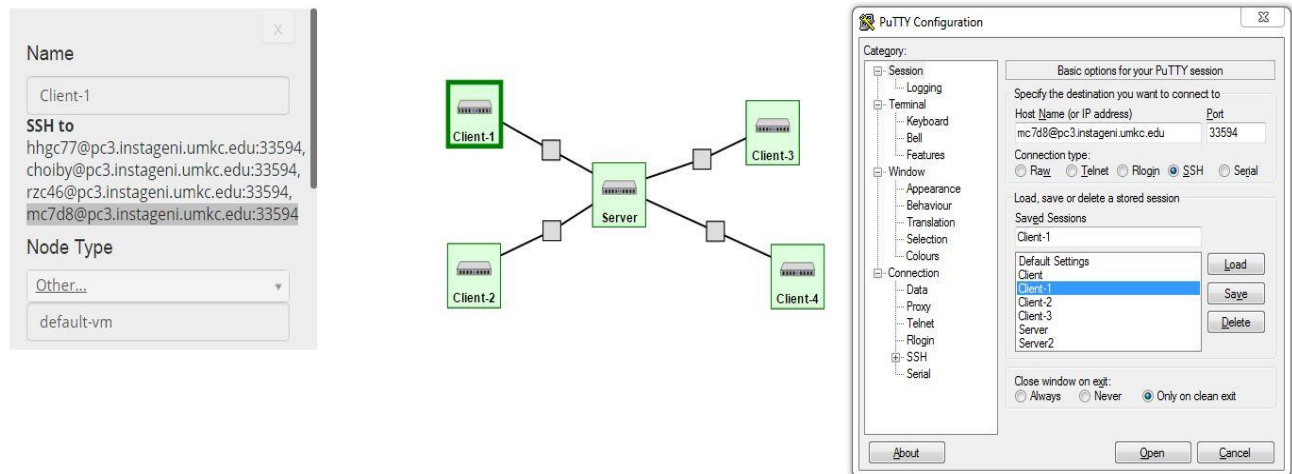
Load Save Delete

Close window on exit:

☐ Always ☐ Never ☒ Only on clean exit

About Open Cancel

Opening Server in Putty by giving Valid Details



Opening Client in Putty by giving Valid Details

3. Project Description

Need to develop a simple chat program (like google hangout and skype chat).

3.1 Question-1

A chat server will accept a single client connection and display everything the client types. If the client user types 'exit', both client and server will end the program.

Solution:

In this query connection is built between Client and Server, messages are sent from Client to Server and vice-versa. Once the Client enters 'exit' both Server and Client exits from Chat Application. Java program is written for each Client (**Client2ServerPartA.java**) and Server (**ServerPartA.java**) through which messages are received and sent back.

Once the java files are compiled and the class file is executed, Client and Server connections are started as shown below. Once chat between Client and Server is completed, Client sends 'exit' which is received by the Server and the connection between Client and Server gets terminated.

When ServerPartA java file is executed the Server, connection is started similarly Client2ServerPartA is executed to start Client. When Client Enters Message the message is sent to Server and the message entered in Server is sent back to Client as shown below. Finally, when Client exits, Server also exits from Chat.


```

mc7d8@server:~$ vi ServerPartA.java
mc7d8@server:~$ javac ServerPartA.java
mc7d8@server:~$ ls
ServerPartA.class  ServerPartA.java
mc7d8@server:~$ java ServerPartA
Server Started Running
*****

mc7d8@client-1:~$ vi Client2ServerPartA.java
mc7d8@client-1:~$ javac Client2ServerPartA.java
mc7d8@client-1:~$ ls
Client2ServerPartA.class  Client2ServerPartA.java
mc7d8@client-1:~$ java Client2ServerPartA
Client Started Running
*****
Enter Message (Client):

```

Client and Server Connection Started

```

mc7d8@client-1:~$ vi Client2ServerPartA.java
mc7d8@client-1:~$ javac Client2ServerPartA.java
mc7d8@client-1:~$ ls
Client2ServerPartA.class  Client2ServerPartA.java
mc7d8@client-1:~$ java Client2ServerPartA
Client Started Running
*****
Enter Message (Client):
Hello From Client
Message Recieved From Server : Hello From Server
Enter Message (Client):
Hi, How are you??
Message Recieved From Server : Hi Client, I am fine... How about you
Enter Message (Client):
Ok.. Got Some work Catch you Later. Bye
Message Recieved From Server : Ok
Enter Message (Client):

mc7d8@server:~$ vi ServerPartA.java
mc7d8@server:~$ javac ServerPartA.java
mc7d8@server:~$ ls
ServerPartA.class  ServerPartA.java
mc7d8@server:~$ java ServerPartA
Server Started Running
*****
Message Received From Client : Hello From Client
Enter Message (Server):
Hello From Server
Message Received From Client : Hi, How are you??
Enter Message (Server):
Hi Client, I am fine... How about you
Message Received From Client : Ok.. Got Some work Catch you Later. Bye
Enter Message (Server):
Ok

```

Chat Communication between Client and Server

```

mc7d8@client-1:~$ vi Client2ServerPartA.java
mc7d8@client-1:~$ javac Client2ServerPartA.java
mc7d8@client-1:~$ ls
Client2ServerPartA.class  Client2ServerPartA.java
mc7d8@client-1:~$ java Client2ServerPartA
Client Started Running
*****
Enter Message (Client):
Hello From Client
Message Recieved From Server : Hello From Server
Enter Message (Client):
Hi, How are you??
Message Recieved From Server : Hi Client, I am fine... How about you
Enter Message (Client):
Ok.. Got Some work Catch you Later. Bye
Message Recieved From Server : Ok
Enter Message (Client):
exit
Message Recieved From Server : exit
mc7d8@client-1:~$

mc7d8@server:~$ vi ServerPartA.java
mc7d8@server:~$ javac ServerPartA.java
mc7d8@server:~$ ls
ServerPartA.class  ServerPartA.java
mc7d8@server:~$ java ServerPartA
Server Started Running
*****
Message Received From Client : Hello From Client
Enter Message (Server):
Hello From Server
Message Received From Client : Hi, How are you??
Enter Message (Server):
Hi Client, I am fine... How about you
Message Received From Client : Ok.. Got Some work Catch you Later. Bye
Enter Message (Server):
Ok
Message Received From Client : exit
mc7d8@server:~$

```

Client and Server exits from Chat

3.2 Question-2

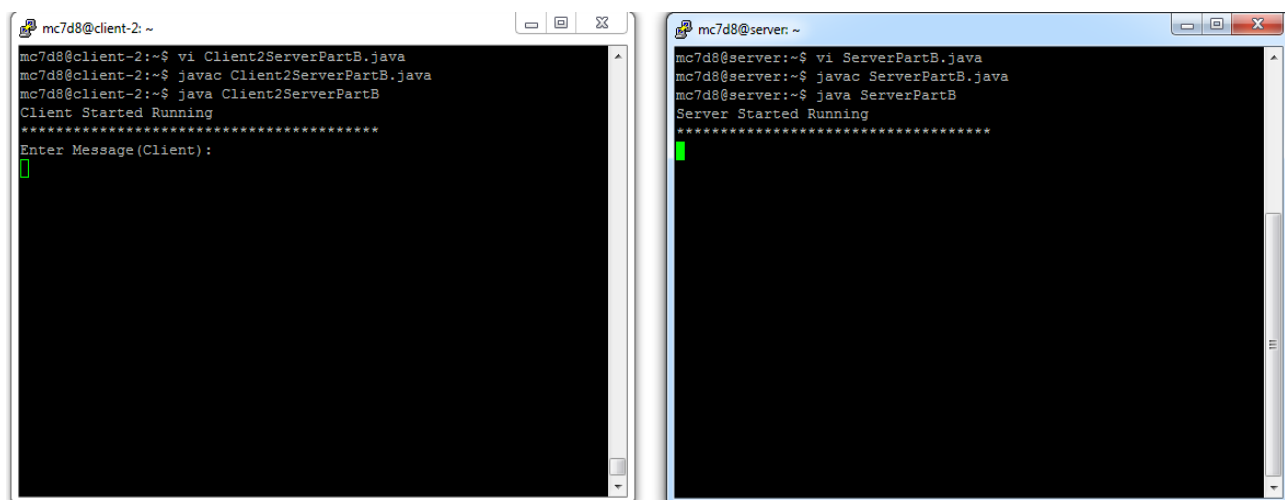
A server now remains 'open' for additional connection once a client quits. The server can handle at most one connection at a time.

Solution:

In this query connection is built between Client and Server, messages are sent from Client to Server and vice-versa. When one of the Client terminates its connection with the Server, the Server waits for the connection from the Clients remaining. At a given time, only one of the Clients can communicate with the Server.

Java program is written for each Client (**Client2ServerPartB.java**) and Server (**ServerPartB.java**) through which messages are received and sent back. Once the java files are compiled and the class file is executed, Client and Server connections are started as shown below. Once chat between Client and Server is completed, Client sends 'exit' which ends Client connection, but Server connection remains and waits for another Client to join.

1. When ServerPartB java file is executed in Putty the Server, connection is started, similarly Client2ServerPartB is executed to start Client.



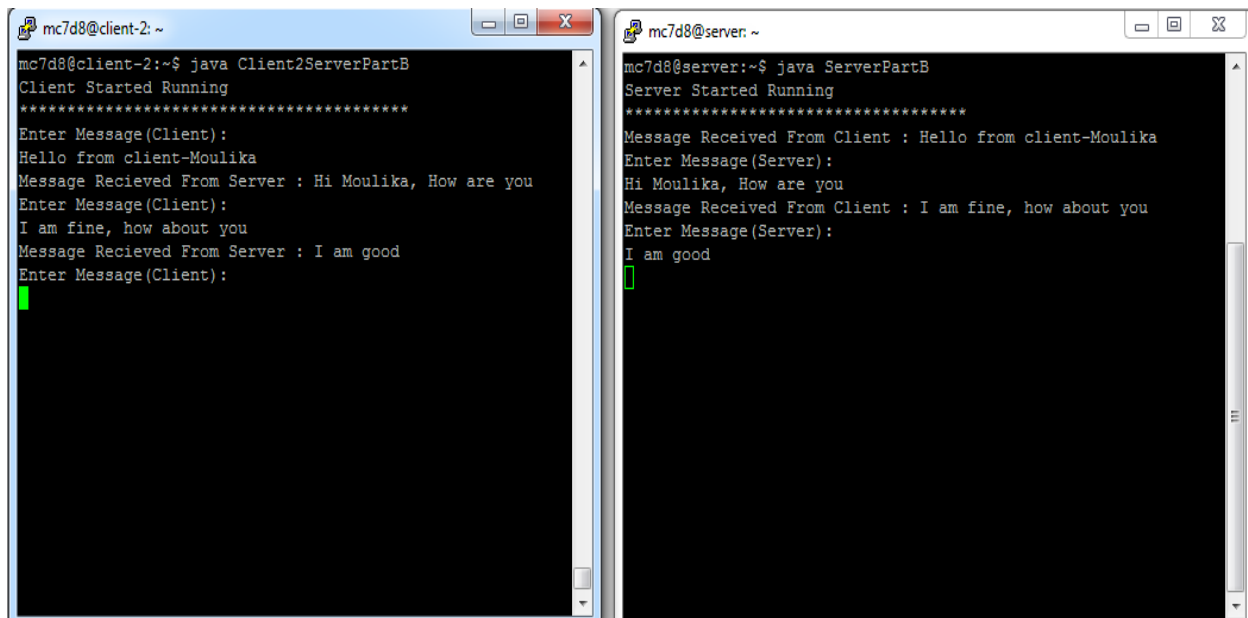
The image shows two terminal windows side-by-side. The left window is titled 'mc7d8@client-2: ~' and shows the following commands and output:
mc7d8@client-2:~\$ vi Client2ServerPartB.java
mc7d8@client-2:~\$ javac Client2ServerPartB.java
mc7d8@client-2:~\$ java Client2ServerPartB
Client Started Running

Enter Message(Client):
[A green cursor is visible on the line following the prompt.]
The right window is titled 'mc7d8@server: ~' and shows the following commands and output:
mc7d8@server:~\$ vi ServerPartB.java
mc7d8@server:~\$ javac ServerPartB.java
mc7d8@server:~\$ java ServerPartB
Server Started Running

[A green cursor is visible on the line following the output.]

Client and Server Connection Started

- When Client Enters Message the message is sent to Server and the message entered in Server is sent back to Client as shown below.



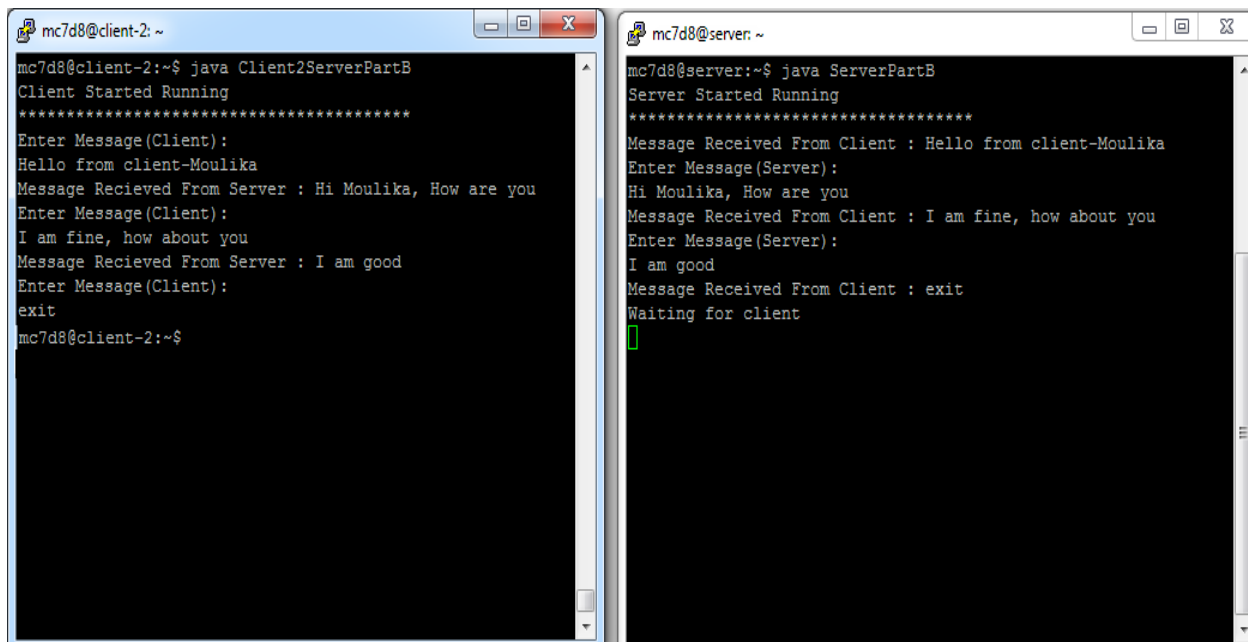
The image shows two terminal windows side-by-side. The left window is titled 'mc7d8@client-2: ~' and the right window is titled 'mc7d8@server: ~'. Both windows show the execution of Java programs for a client-server chat application. The client program is 'Client2ServerPartB' and the server program is 'ServerPartB'. The client sends messages like 'Hello from client-Moulika' and 'I am fine, how about you', which are received by the server. The server responds with 'Hi Moulika, How are you' and 'I am good', which are received by the client. The client then enters 'exit'.

```
mc7d8@client-2:~$ java Client2ServerPartB
Client Started Running
*****
Enter Message(Client):
Hello from client-Moulika
Message Recieved From Server : Hi Moulika, How are you
Enter Message(Client):
I am fine, how about you
Message Recieved From Server : I am good
Enter Message(Client):
[ ]

mc7d8@server:~$ java ServerPartB
Server Started Running
*****
Message Received From Client : Hello from client-Moulika
Enter Message(Server):
Hi Moulika, How are you
Message Received From Client : I am fine, how about you
Enter Message(Server):
I am good
[ ]
```

Chat Communication between Client and Server

- Finally, when Client exits, Server Waits for another Client to join and start the communication



The image shows two terminal windows side-by-side. The left window is titled 'mc7d8@client-2: ~' and the right window is titled 'mc7d8@server: ~'. Both windows show the execution of Java programs for a client-server chat application. The client program is 'Client2ServerPartB' and the server program is 'ServerPartB'. The client sends messages like 'Hello from client-Moulika' and 'I am fine, how about you', which are received by the server. The server responds with 'Hi Moulika, How are you' and 'I am good', which are received by the client. The client then enters 'exit'. The server then displays 'Waiting for client'.

```
mc7d8@client-2:~$ java Client2ServerPartB
Client Started Running
*****
Enter Message(Client):
Hello from client-Moulika
Message Recieved From Server : Hi Moulika, How are you
Enter Message(Client):
I am fine, how about you
Message Recieved From Server : I am good
Enter Message(Client):
exit
mc7d8@client-2:~$

mc7d8@server:~$ java ServerPartB
Server Started Running
*****
Message Received From Client : Hello from client-Moulika
Enter Message(Server):
Hi Moulika, How are you
Message Received From Client : I am fine, how about you
Enter Message(Server):
I am good
Message Received From Client : exit
Waiting for client
[ ]
```

Client exists but Server waits for Client

4. When Client-2 connection is started, it starts communication with Server

```

mc7d8@client-2: ~
mc7d8@client-2:~$ java Client2ServerPartB
Client Started Running
*****
Enter Message(Client):
Hello from client-Moulika
Message Recieved From Server : Hi Moulika, How are you
Enter Message(Client):
I am fine, how about you
Message Recieved From Server : I am good
Enter Message(Client):
exit
Message Recieved From Server : exit
mc7d8@client-2:~$

mc7d8@client-3: ~
mc7d8@client-3:~$ java Client2ServerPartB
Client Started Running
*****
Enter Message(Client):
Hello from client-Anusha
Message Recieved From Server : Hi Anusha, How are you
Enter Message(Client):
I am fine dear Server
Message Recieved From Server : Howz your work
Enter Message(Client):
Great
mc7d8@client-3:~$

mc7d8@server: ~
mc7d8@server:~$ java ServerPartB
Server Started Running
*****
Message Received From Client : Hello from client-Moulika
Enter Message(Server):
Hi Moulika, How are you
Message Received From Client : I am fine, how about you
Enter Message(Server):
I am good
Message Received From Client : exit
Waiting for client
Message Received From Client : Hello from client-Anusha
Enter Message(Server):
Hi Anusha, How are you
Message Received From Client : I am fine dear Server
Enter Message(Server):
Howz your work
Message Received From Client : Great
Enter Message(Server):

```

Client 2 Enters Chat and communicate with Server

```

mc7d8@client-2: ~
mc7d8@client-2:~$ java Client2ServerPartB
Client Started Running
*****
Enter Message(Client):
Hello from client-Moulika
Message Recieved From Server : Hi Moulika, How are you
Enter Message(Client):
I am fine, how about you
Message Recieved From Server : I am good
Enter Message(Client):
exit
Message Recieved From Server : exit
mc7d8@client-2:~$

mc7d8@client-3: ~
mc7d8@client-3:~$ java Client2ServerPartB
Client Started Running
*****
Enter Message(Client):
Hello from client-Anusha
Message Recieved From Server : Hi Anusha, How are you
Enter Message(Client):
I am fine dear Server
Message Recieved From Server : Howz your work
Enter Message(Client):
Great
Message Recieved From Server : Good
Enter Message(Client):
exit
mc7d8@client-3:~$

mc7d8@server: ~
mc7d8@server:~$ java ServerPartB
Server Started Running
*****
Message Received From Client : Hello from client-Moulika
Enter Message(Server):
Hi Moulika, How are you
Message Received From Client : I am fine, how about you
Enter Message(Server):
I am good
Message Received From Client : exit
Waiting for client
Message Received From Client : Hello from client-Anusha
Enter Message(Server):
Hi Anusha, How are you
Message Received From Client : I am fine dear Server
Enter Message(Server):
Howz your work
Message Received From Client : Great
Enter Message(Server):
Good
Waiting for client

```

Client 2 exits from Chat

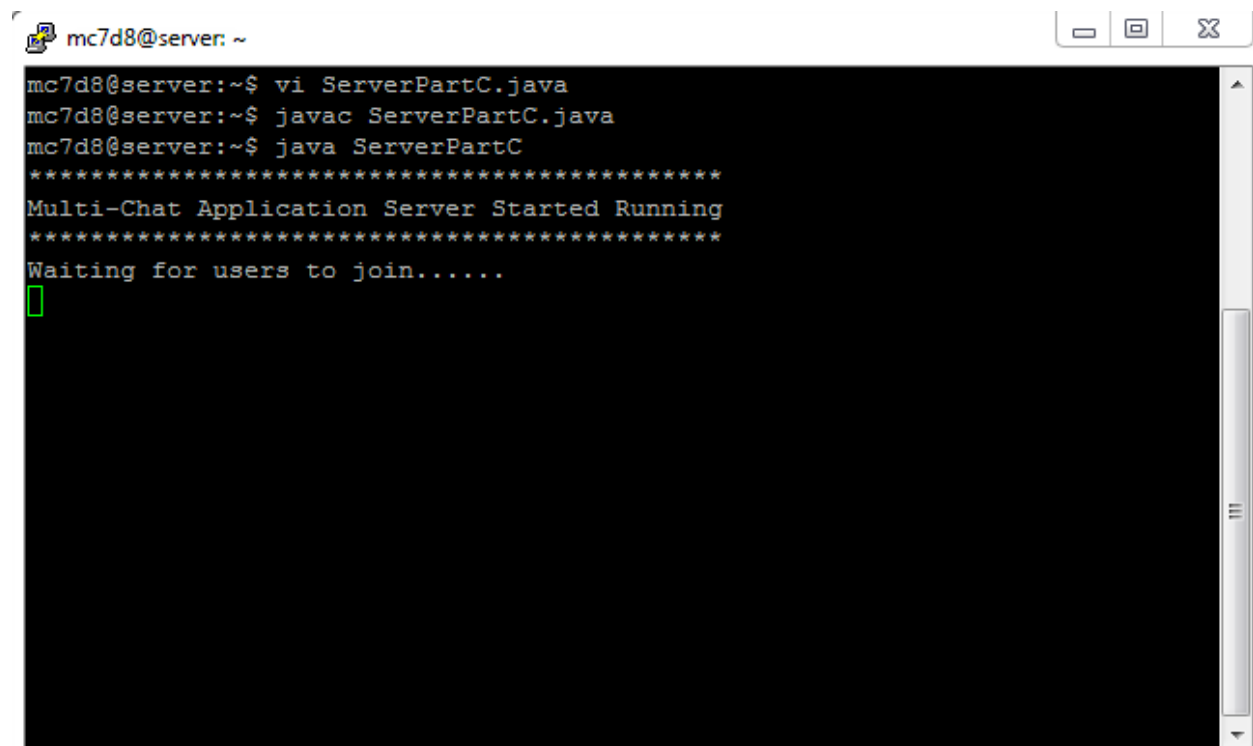
3.3 Question-3

A server now can handle multiple clients at the same time. The output from all the connected clients will appear on the server's screen.

Solution:

In this query connection is built between Client and Server, messages are sent from Client to Server and vice-versa. Java files are created for each of the Clients and Server through which messages are received and sent back. In this query, we illustrate that the Server can handle any number of Clients at the same time. As the Clients send the messages to the Server, they appear on the display screen of the Server.

1. Initially ServerPartC java file is executed, which waits for clients to join

A terminal window titled 'mc7d8@server: ~' with standard window controls. The terminal shows the following commands and output:

```
mc7d8@server:~$ vi ServerPartC.java
mc7d8@server:~$ javac ServerPartC.java
mc7d8@server:~$ java ServerPartC
*****
Multi-Chat Application Server Started Running
*****
Waiting for users to join.....
█
```

Server Starts and waits for clients

2. Next Client-1 is started, where the name of Client is entered, so that it will be easy to differentiate between clients. Here the name is entered as 'Moulika' and the chat entered in Client-1 is displayed on Server screen with Name of Client <Moulika>

```

mc7d8@server: ~
mc7d8@server:~$ vi ServerPartC.java
mc7d8@server:~$ javac ServerPartC.java
mc7d8@server:~$ java ServerPartC
*****
Multi-Chat Application Server Started Running
*****
Waiting for users to join.....
█

mc7d8@client-1: ~
mc7d8@client-1:~$ java Client2ServerPartC
Hi User..Enter your name:
Moulika
Hello Moulika ,welcome to our Multi-Chat Application
█

```

Client-1 is added into Chat

```

mc7d8@server: ~
mc7d8@server:~$ vi ServerPartC.java
mc7d8@server:~$ javac ServerPartC.java
mc7d8@server:~$ java ServerPartC
*****
Multi-Chat Application Server Started Running
*****
Waiting for users to join.....
<Moulika> Hello, Hi How are you
<Moulika> Howz you day today
█

mc7d8@client-1: ~
mc7d8@client-1:~$ java Client2ServerPartC
Hi User..Enter your name:
Moulika
Hello Moulika ,welcome to our Multi-Chat Application
Hello, Hi How are you
Howz you day today
█

```

Client-1 Started Communicating with Server

- When Client-2 in similar way as Client-1 is added. Here Client-2 name is entered is Nikitha and the chat entered in Client-1 and Client-2 are displayed on Server.

```

mc7d8@server: ~
mc7d8@server:~$ vi ServerPartC.java
mc7d8@server:~$ javac ServerPartC.java
mc7d8@server:~$ java ServerPartC
*****
Multi-Chat Application Server Started Running
*****
Waiting for users to join.....
<Moulika> Hello, Hi How are you
<Moulika> Howz you day today
<Nikitha> Hello,Good Morning
<Moulika> Good Morning, I am travelling to California
<Nikitha> I am travelling to Florida
█

mc7d8@client-1: ~
mc7d8@client-1:~$ java Client2ServerPartC
Hi User..Enter your name:
Moulika
Hello Moulika ,welcome to our Multi-Chat Application
Hello, Hi How are you
Howz you day today
Good Morning, I am travelling to California
█

mc7d8@client-2: ~
mc7d8@client-2:~$ java Client2ServerPartC
Hi User..Enter your name:
Nikitha
Hello Nikitha ,welcome to our Multi-Chat Application
Hello,Good Morning
I am travelling to Florida
█

```

Client-2 is started and chat is displayed on Server

- Once the client enters /exit then the client will be exited from Chat and the intimation is given to Server. Here Client-2 has exited from chat and the response is displayed on Server

```

mc7d8@server: ~
mc7d8@server:~$ vi ServerPartC.java
mc7d8@server:~$ javac ServerPartC.java
mc7d8@server:~$ java ServerPartC
*****
Multi-Chat Application Server Started Running
*****
Waiting for users to join.....
<Moulika> Hello, Hi How are you
<Moulika> Howz you day today
<Nikitha> Hello,Good Morning
<Moulika> Good Morning, I am travelling to California
<Nikitha> I am travelling to Florida
<Nikitha> /exit
Nikitha exited from chat
█

mc7d8@client-1: ~
mc7d8@client-1:~$ java Client2ServerPartC
Hi User..Enter your name:
Moulika
Hello Moulika ,welcome to our Multi-Chat Application
Hello, Hi How are you
Howz you day today
Good Morning, I am travelling to California
█

mc7d8@client-2: ~
mc7d8@client-2:~$ java Client2ServerPartC
Hi User..Enter your name:
Nikitha
Hello Nikitha ,welcome to our Multi-Chat Application
Hello,Good Morning
I am travelling to Florida
/exit
Nikitha left from Chat
mc7d8@client-2:~$ █

```

Client-2 exited from Chat

3.4 Question-4

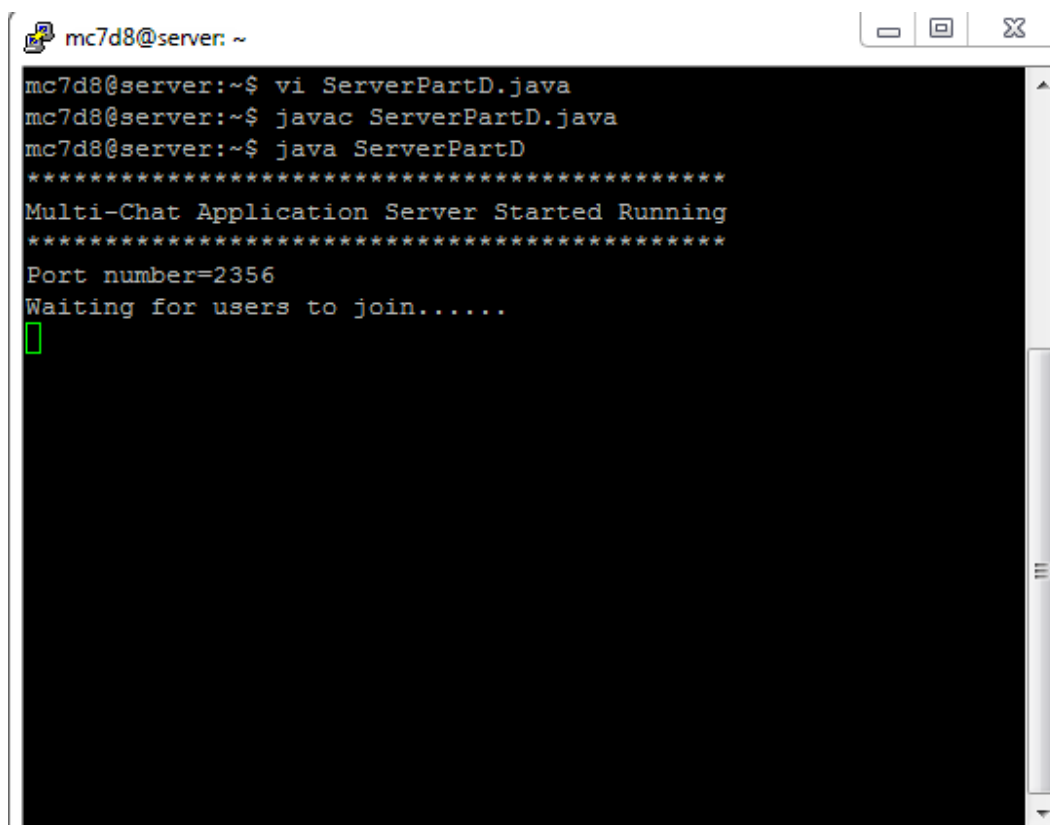
A server now echoes all the text received from any of the connected clients to all.

Solution:

Java files are created for each of the Clients and Server through which messages are received and sent back. In this query, we show that when any of the Clients sends its message to the Server, it broadcasts the message to the remaining Clients with the name of the Client specified.

Here we are connecting to 4 Clients, whenever new Client is started to Chat Application the intimation is given to all other Clients. Also, when Client is left from Chat the exit message is sent to Server. The communication between client and server is echoed to all other servers. For example, Client-1 sends message 'hi', this message will be displayed on Client-2, Client-3, Client-4 along with Server.

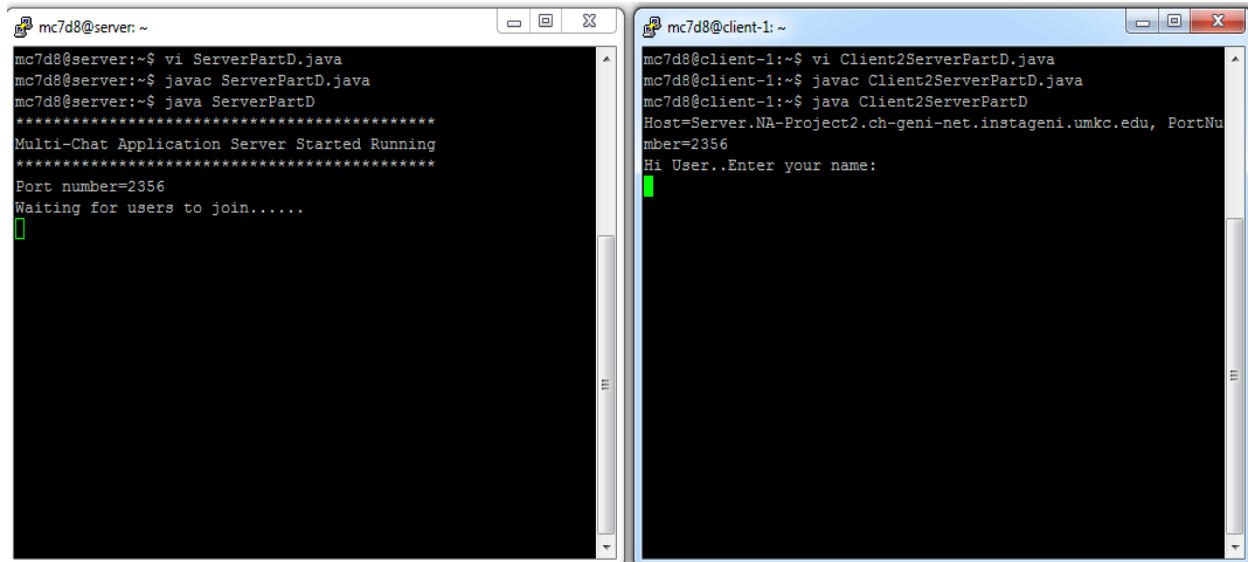
1. Initially ServerPartD java file is executed, which waits for clients to join

A terminal window titled 'mc7d8@server: ~' with standard window controls. The terminal shows the following commands and output:

```
mc7d8@server:~$ vi ServerPartD.java
mc7d8@server:~$ javac ServerPartD.java
mc7d8@server:~$ java ServerPartD
*****
Multi-Chat Application Server Started Running
*****
Port number=2356
Waiting for users to join.....
█
```

Server Starts and waits for clients

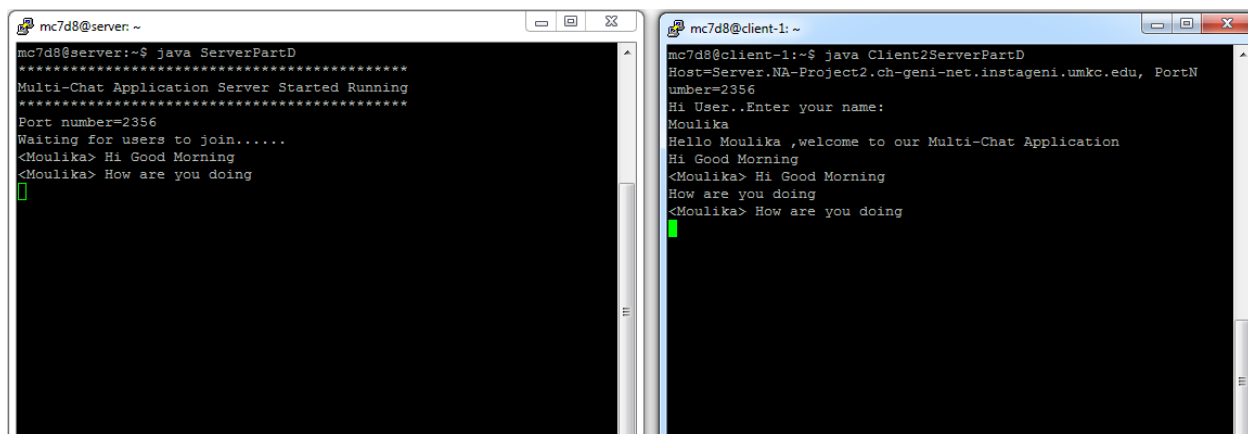
- Next Client-1 is started, where the name of Client is entered, so that it will be easy to differentiate between clients. Here the name is entered as 'Moulika' and the chat entered in Client-1 is displayed on Server screen with Name of Client <Moulika>



```
mc7d8@server:~$ vi ServerPartD.java
mc7d8@server:~$ javac ServerPartD.java
mc7d8@server:~$ java ServerPartD
*****
Multi-Chat Application Server Started Running
*****
Port number=2356
Waiting for users to join.....
█

mc7d8@client-1:~$ vi Client2ServerPartD.java
mc7d8@client-1:~$ javac Client2ServerPartD.java
mc7d8@client-1:~$ java Client2ServerPartD
Host=Server.NA-Project2.ch-geni-net.instageni.umkc.edu, PortNumber=2356
Hi User..Enter your name:
█
```

Client-1 is added into Chat



```
mc7d8@server:~$ java ServerPartD
*****
Multi-Chat Application Server Started Running
*****
Port number=2356
Waiting for users to join.....
<Moulika> Hi Good Morning
<Moulika> How are you doing
█

mc7d8@client-1:~$ java Client2ServerPartD
Host=Server.NA-Project2.ch-geni-net.instageni.umkc.edu, PortNumber=2356
Hi User..Enter your name:
Moulika
Hello Moulika ,welcome to our Multi-Chat Application
Hi Good Morning
<Moulika> Hi Good Morning
How are you doing
<Moulika> How are you doing
█
```

Client-1 Started Communicating with Server

- When Client-2 is added then the New user added notification is sent to Client-1, so that they can communicate each other. Here Client-2 name is entered is Nikitha and the chat communication between Client-1 and Client-2 is displayed on Server.

```

mc7d8@server: ~
mc7d8@server:~$ java ServerPartD
*****
Multi-Chat Application Server Started Running
*****
Port number=2356
Waiting for users to join.....
<Moulika> Hi Good Morning
<Moulika> How are you doing
█

mc7d8@client-1: ~
mc7d8@client-1:~$ java Client2ServerPartD
Host=Server.NA-Project2.ch-geni-net.instageni.umkc.edu, PortN
umber=2356
Hi User..Enter your name:
Moulika
Hello Moulika ,welcome to our Multi-Chat Application
Hi Good Morning
<Moulika> Hi Good Morning
How are you doing
<Moulika> How are you doing
New User Nikitha joined Multi-Chat Application
█

mc7d8@client-2: ~
mc7d8@client-2:~$ java Client2ServerPartD
Host=Server.NA-Project2.ch-geni-net.instageni.umkc.edu,
PortNumber=2356
Hi User..Enter your name:
Nikitha
Hello Nikitha ,welcome to our Multi-Chat Application
█

```

Client-2 is added and intimation sent to Client-1

```

mc7d8@server: ~
mc7d8@server:~$ java ServerPartD
*****
Multi-Chat Application Server Started Running
*****
Port number=2356
Waiting for users to join.....
<Moulika> Hi Good Morning
<Moulika> How are you doing
<Nikitha> Hello Friends
<Moulika> Hi Nikitha
<Nikitha> Hi Moulika, How are you
█

mc7d8@client-1: ~
mc7d8@client-1:~$ java Client2ServerPartD
Host=Server.NA-Project2.ch-geni-net.instageni.umkc.edu, PortN
umber=2356
Hi User..Enter your name:
Moulika
Hello Moulika ,welcome to our Multi-Chat Application
Hi Good Morning
<Moulika> Hi Good Morning
How are you doing
<Moulika> How are you doing
New User Nikitha joined Multi-Chat Application
<Nikitha> Hello Friends
Hi Nikitha
<Moulika> Hi Nikitha
<Nikitha> Hi Moulika, How are you
█

mc7d8@client-2: ~
mc7d8@client-2:~$ java Client2ServerPartD
Host=Server.NA-Project2.ch-geni-net.instageni.umkc.edu,
PortNumber=2356
Hi User..Enter your name:
Nikitha
Hello Nikitha ,welcome to our Multi-Chat Application
Hello Friends
<Nikitha> Hello Friends
<Moulika> Hi Nikitha
Hi Moulika, How are you
<Nikitha> Hi Moulika, How are you
█

```

Client-1 and Client-1 are communicating with each other

4. Similarly, other two clients Client-3 and Client-4 are also added into Chat Application. So, four clients group chat and the same is displayed on Server.

Client-3 is added and started communicating with other clients

Client-4 is added and group chat between all client

5. Once the client enters /exit then the client will be exited from Chat and the intimation is given to Server. Here Client-1 has exited from chat and the response is displayed on Server

```

mc7d8@server:~$ vi ServerPartD.java
mc7d8@server:~$ javac ServerPartD.java
mc7d8@server:~$ java ServerPartD
*****
Multi-Chat Application Server Started Running
*****
Port number=2356
Waiting for users to join.....
<Moulika> Hi Good Morning
<Moulika> How are you doing
<Nikitha> Hello friends
<Moulika> Hi Nikitha
<Nikitha> Hi Moulika,How are you
New User Anusha joined Multi-Chat Application
<Anusha> Hi All, Very Good Morning to All
<Nikitha> Good Morning Anusha
<Anusha> Hi Nikitha howz yesterday party
New User HarryPotter joined Multi-Chat Application
Hi friends, can we meet today
<Moulika> Hi friends, can we meet today
<HarryPotter> Ya sure we will
<Anusha> Timings and place please
<Nikitha> At 5PM in Miller Nicholas Library
Moulika left from Chat
Moulika exited from chat

mc7d8@client-1:~$
mc7d8@client-2:~$
mc7d8@client-3:~$
mc7d8@client-4:~$

```

Client-1 Exited from Chat

4. References

<http://portal.geni.net/secure/dashboard.php>

<http://stackoverflow.com/>

<http://www.putty.org/>