11/10/2015

Rajashekhar reddy Golipally - 16207852

Swetha Chandra Karroti - 16208563

Pallavi Ramineni - 16208562



NETWORK ARCHETECTURE 1

PROJECT



**NETWORK ARCHETECTURE 1**

**PROJECT**

**Project Participation**

**Initial Setup:**

**Rajashekhar .G**

* Geni account allocation
* Resource collection
* Related study
* Tux usage and communication.

**Phase 1:**

**Pallavi and Swetha**

* Java code for phase 1
* Initial phase 2 lookup
* Resource lookup and related study

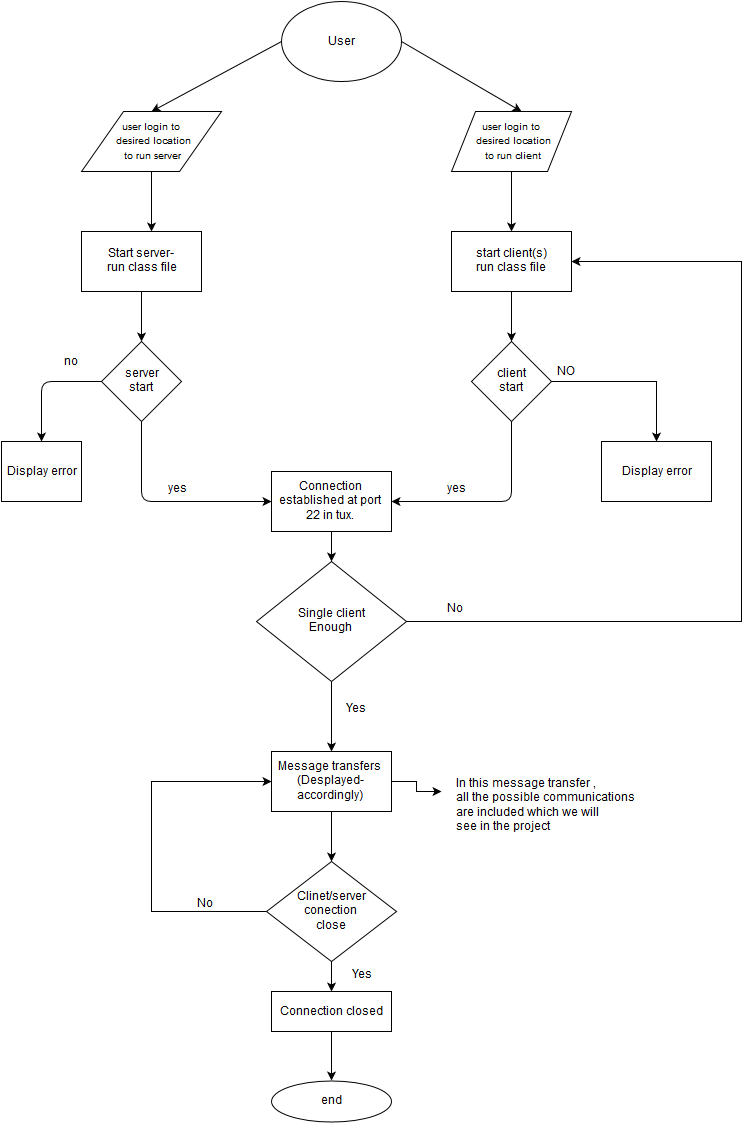
**Phase 2:**

**Whole Team**

* This being difficult part, had to study many similar ideas and after try to apply in our own style.
* Java coding
* Related study and execution
* Final Documentation

**Let’s draw the overall process in a flow chart for better understanding**

**Flow Chart**

****

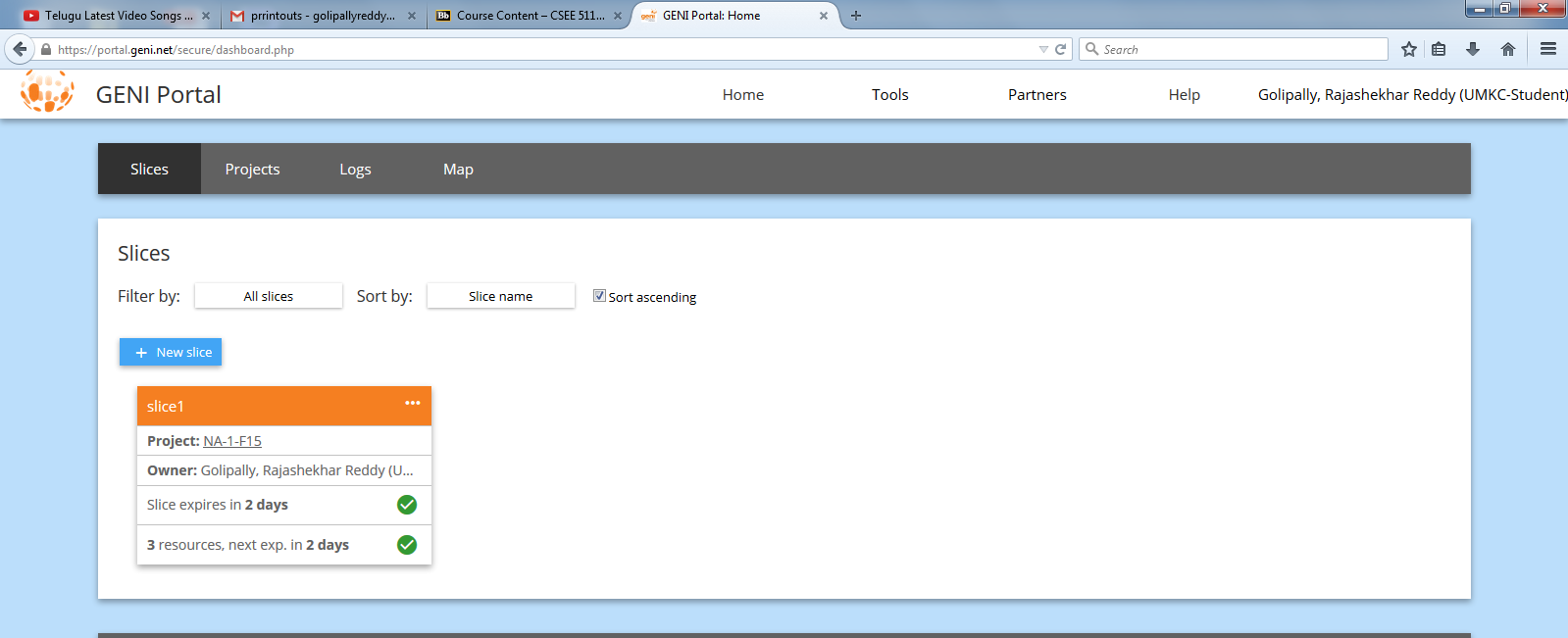
**Requirements:**

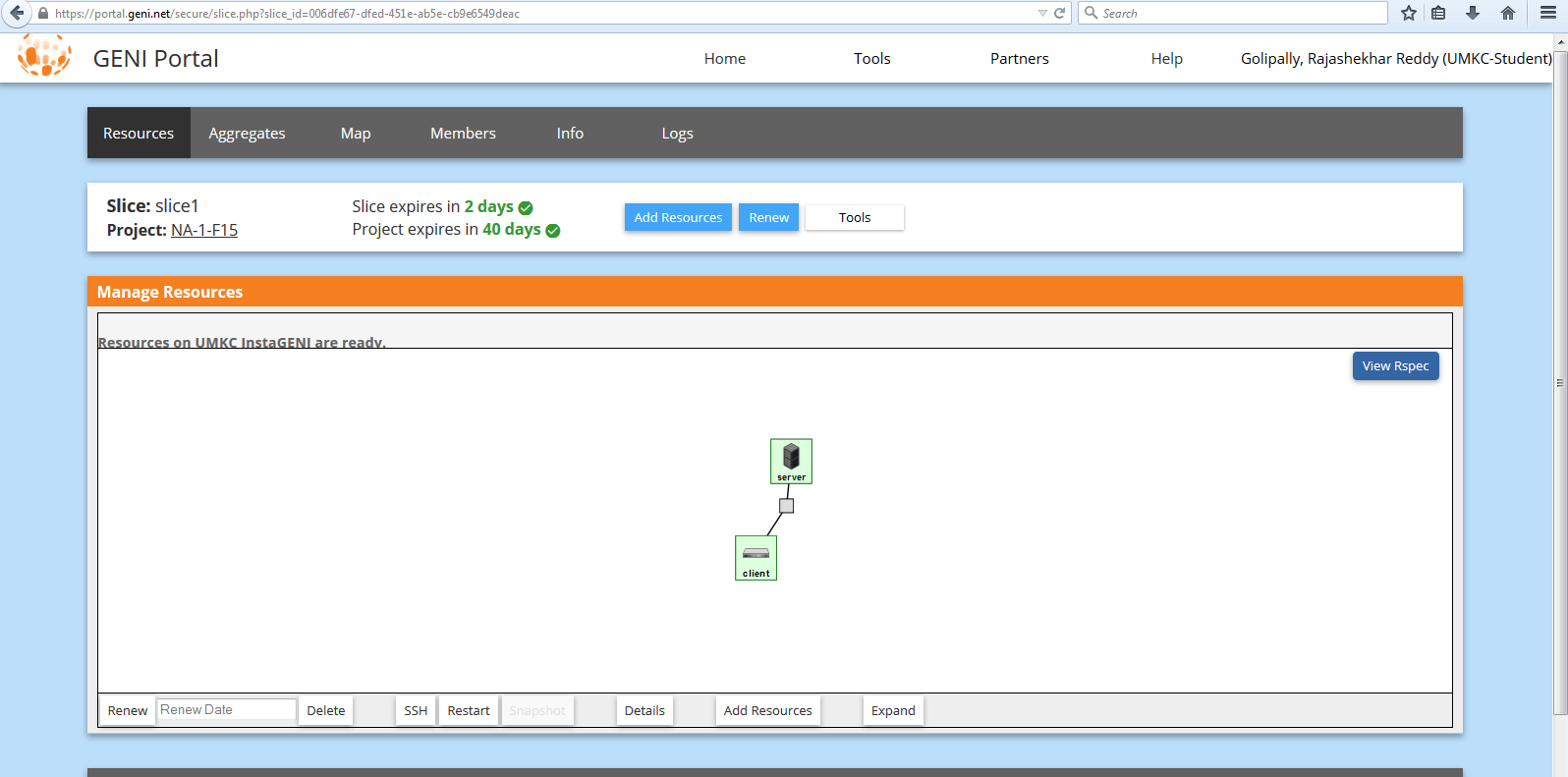
* Account in genie (platform for networking professionals) and nodes generated where we can work on.
* Ssh key pairs to login to nodes and run the scripts
* Code to get the desired outputs, other required hardware

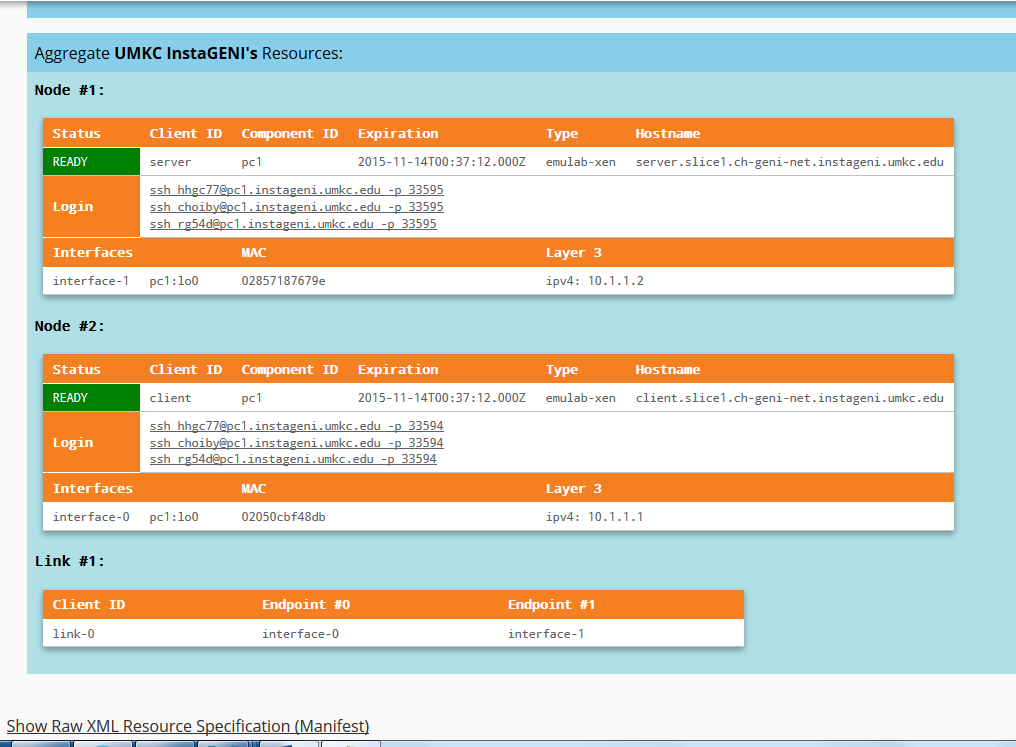
**Initial difficulty:**

There were issue to login to nodes using genie, some issues with passphrase, had to re generate new SSH which takes time to get distributed in mean time in the slice. On advice of TA, we have started work in tux. We logged into tux at college network locations and used tux to run client and chat programs.

**Geni Account:**







**Objective:**

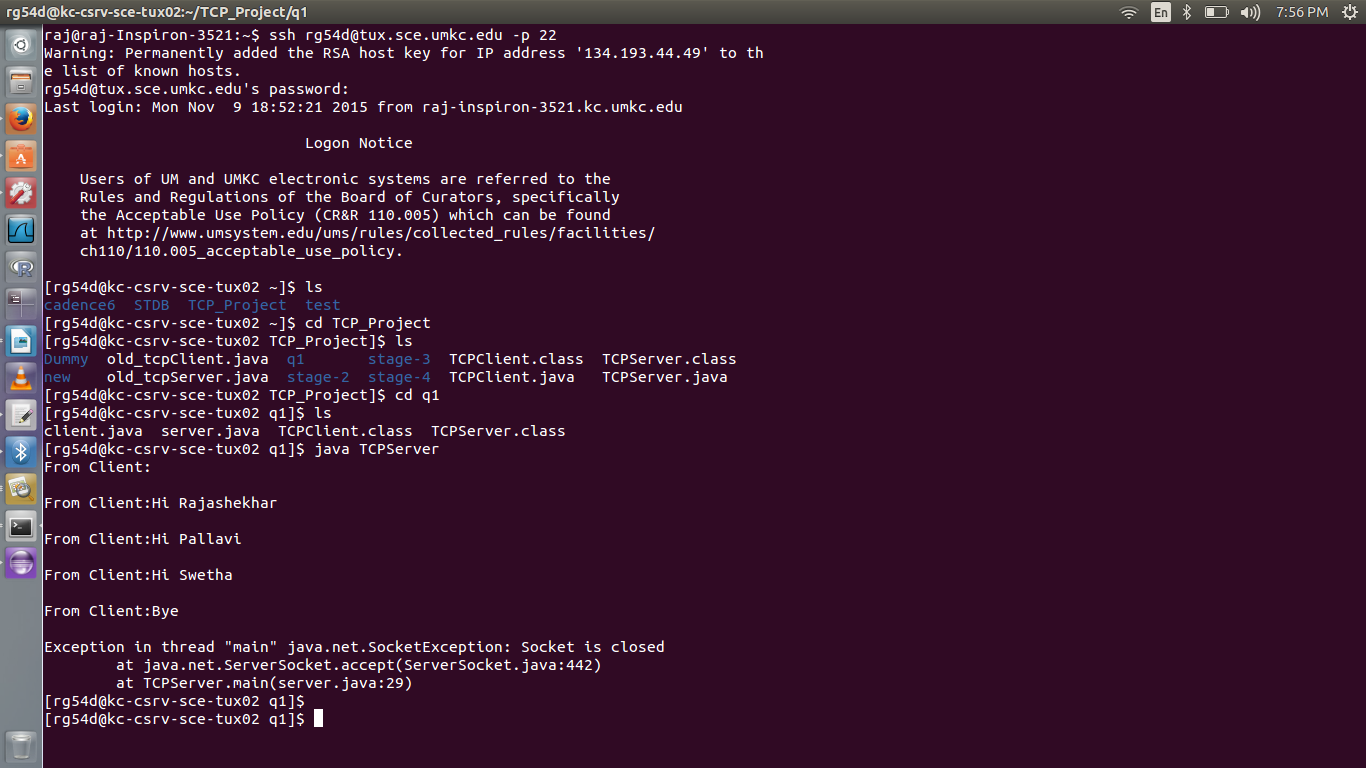
Given project has two things two be done

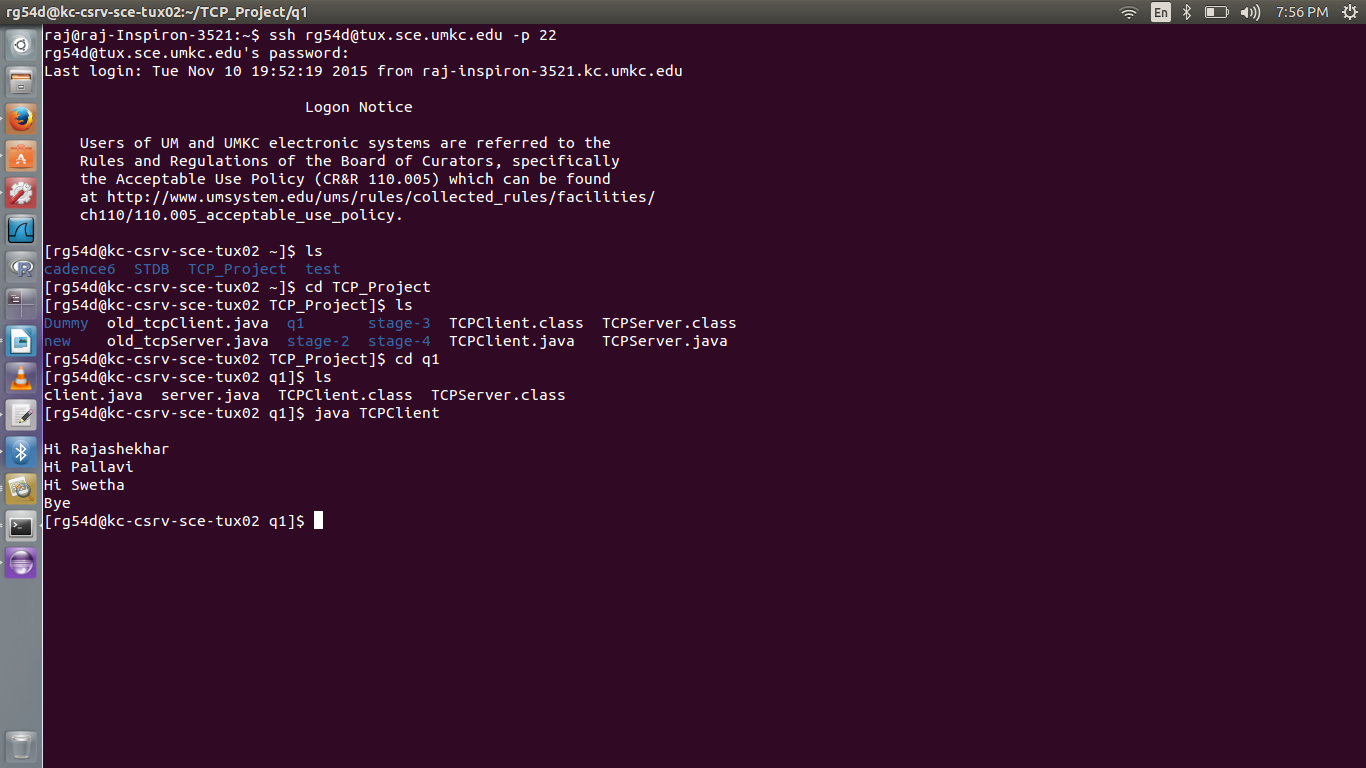
**Phase 1**

Build a server client communication which accepts communication, print whatever client types and which exit when client type **“bye”**

**Execution:**

* Type hi Your names
* Server print in it : hi Your names
* Type bye
* Server and client close the connection!

**Screenshots**

****

**Phase 2**

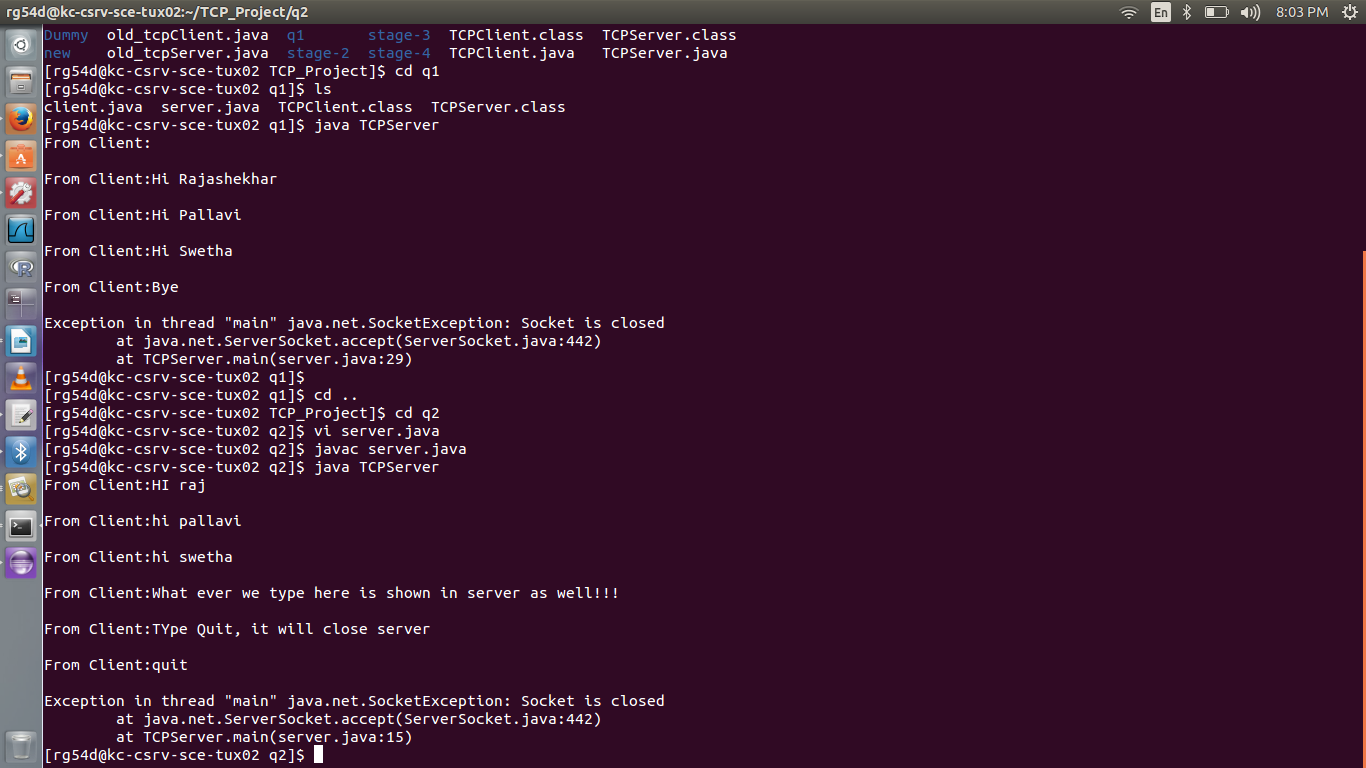
**In phase 2 we have 4 tasks to be done using a multi threat tcp client and server**

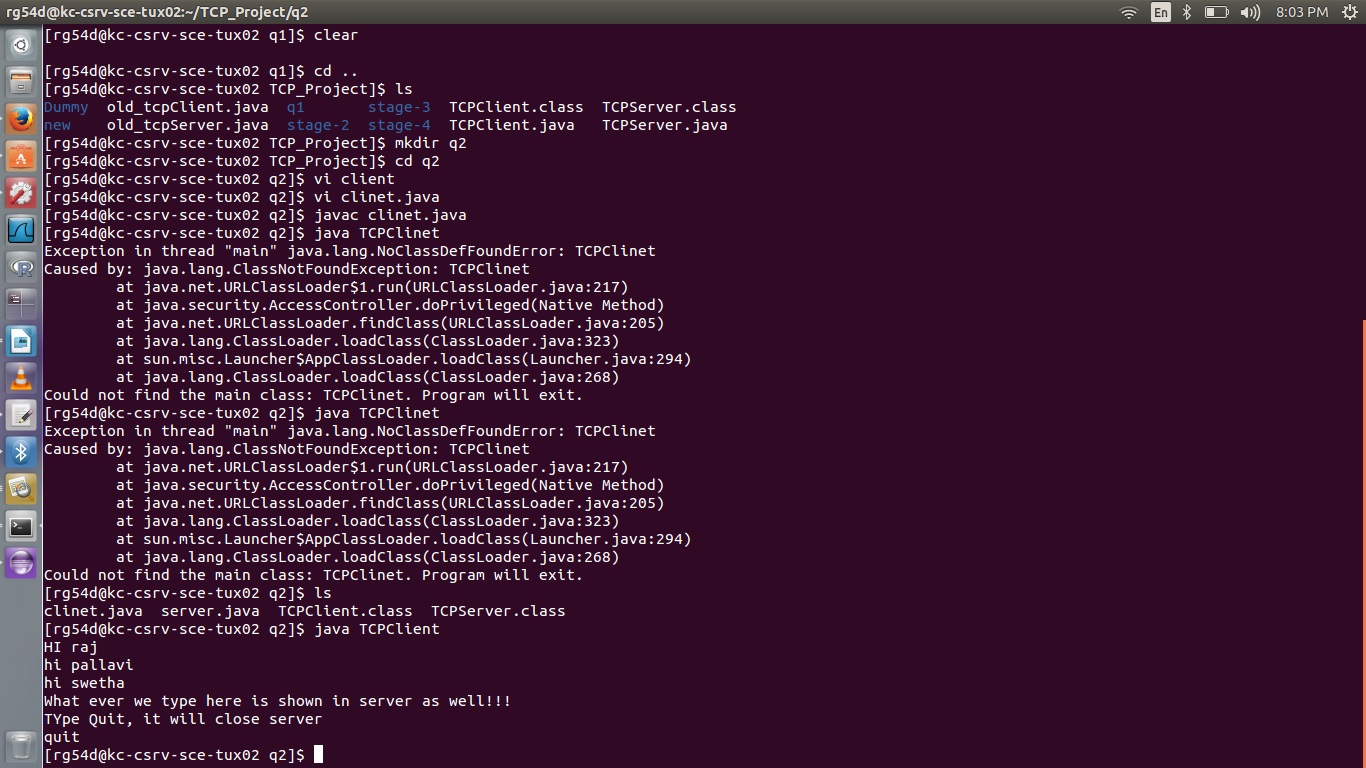
**a) In this Client connects to one server and print whatever client type in the client window, just like phase one, here also when client type “quit” server and client will both quit**

**Execution:**

* Client type HI name
* Server print hi name
* Client type “quit”
* Server and client quit

**Screen Shots**

****

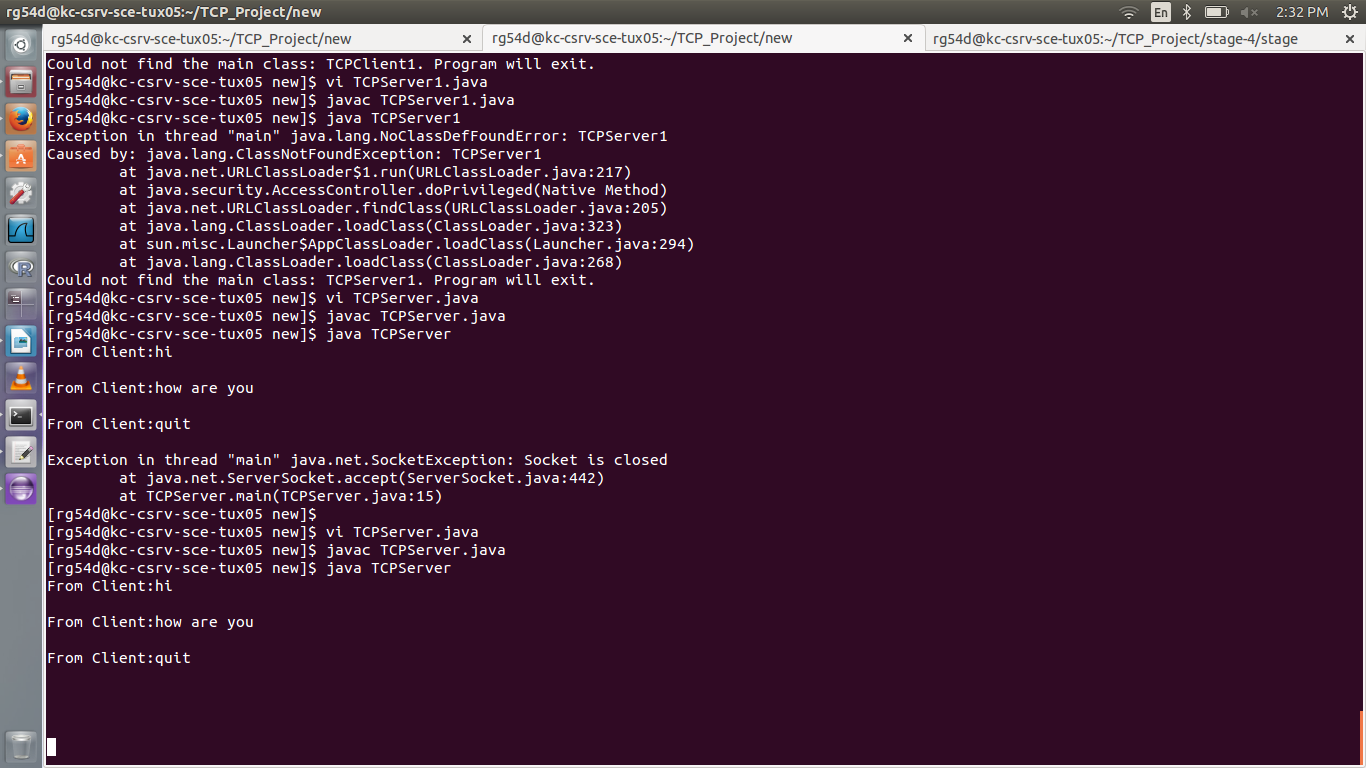
****

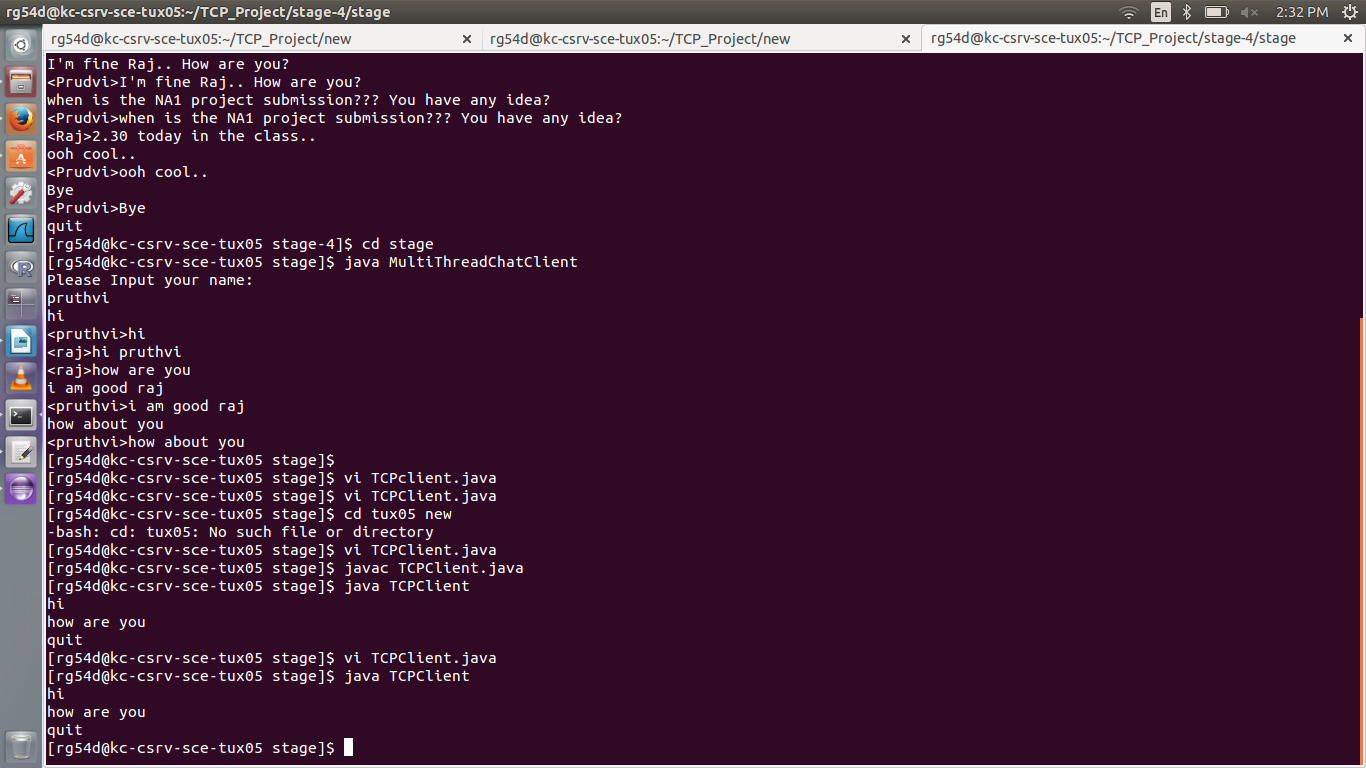
**b) The second part in the phase two is extension of 2a, here after client type quit, the server should still keep listening at the desired port. But the client will stop running**

**Execution:**

* Client type hi
* Server type hi
* Client type bye
* Client connection is closed
* Server keeps looking for connection

**Screen Shots**



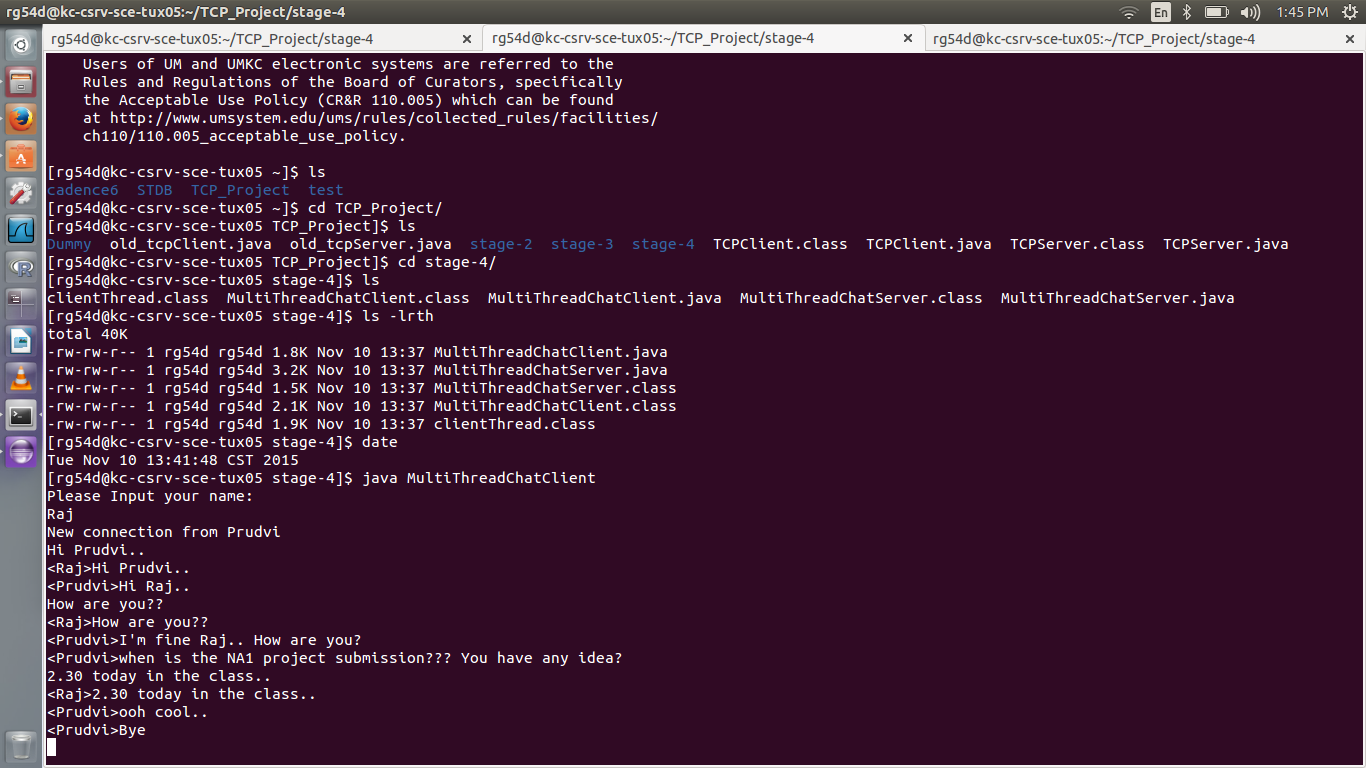


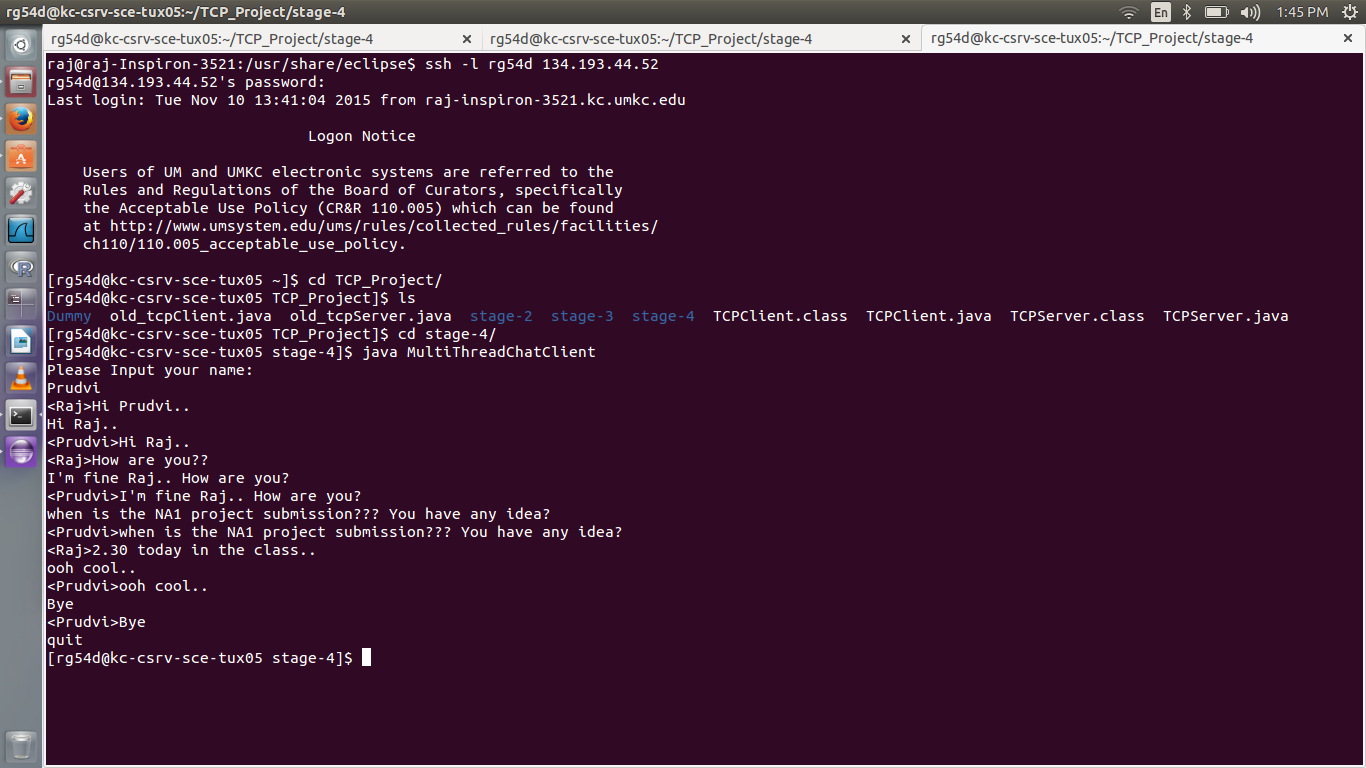
**c) The third phase of the phase 2 is a multi-thread program where we have two clients and one server. No of servers will depend on the no of execution windows we use. Here the client (each) connects to server and whenever new node is connected, it is echoed to all the remaining nodes and outputs are displayed on server screen**

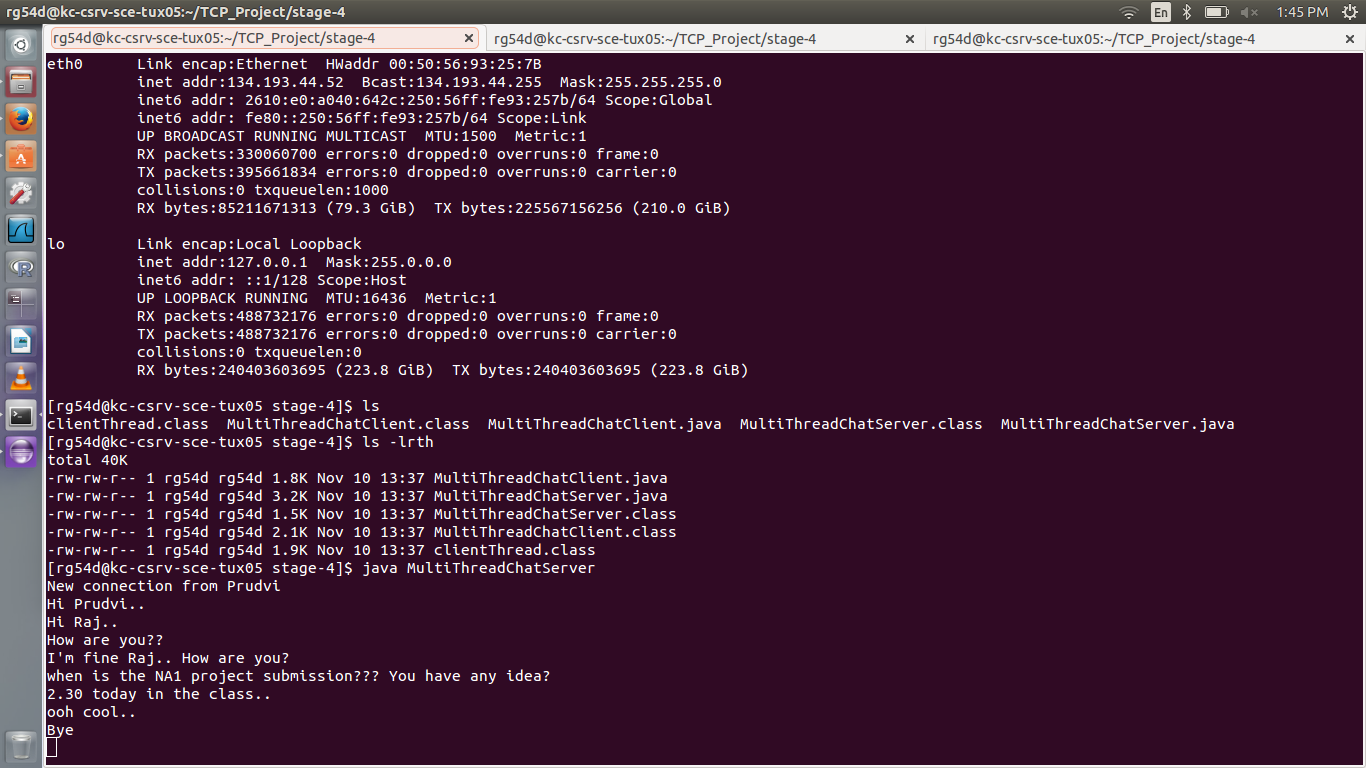
**Execution**

* Server listening
* Msg to client (Enter your name”)
* Client enter name
* Printed in server and other opened nodes
* Second client type name
* Info about client will be sent to both server and other node
* Client communicates
* Printed on server and printed on individual clients

**Screen Shots**





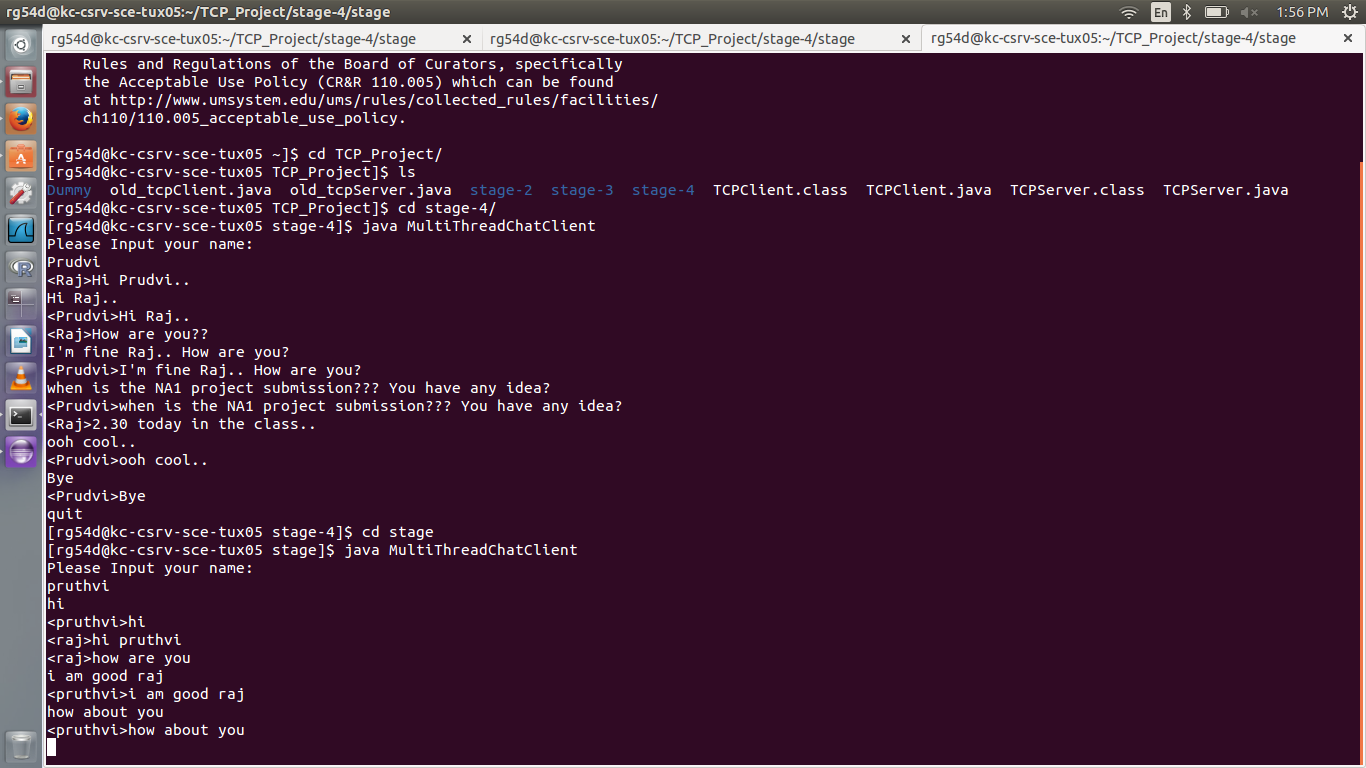


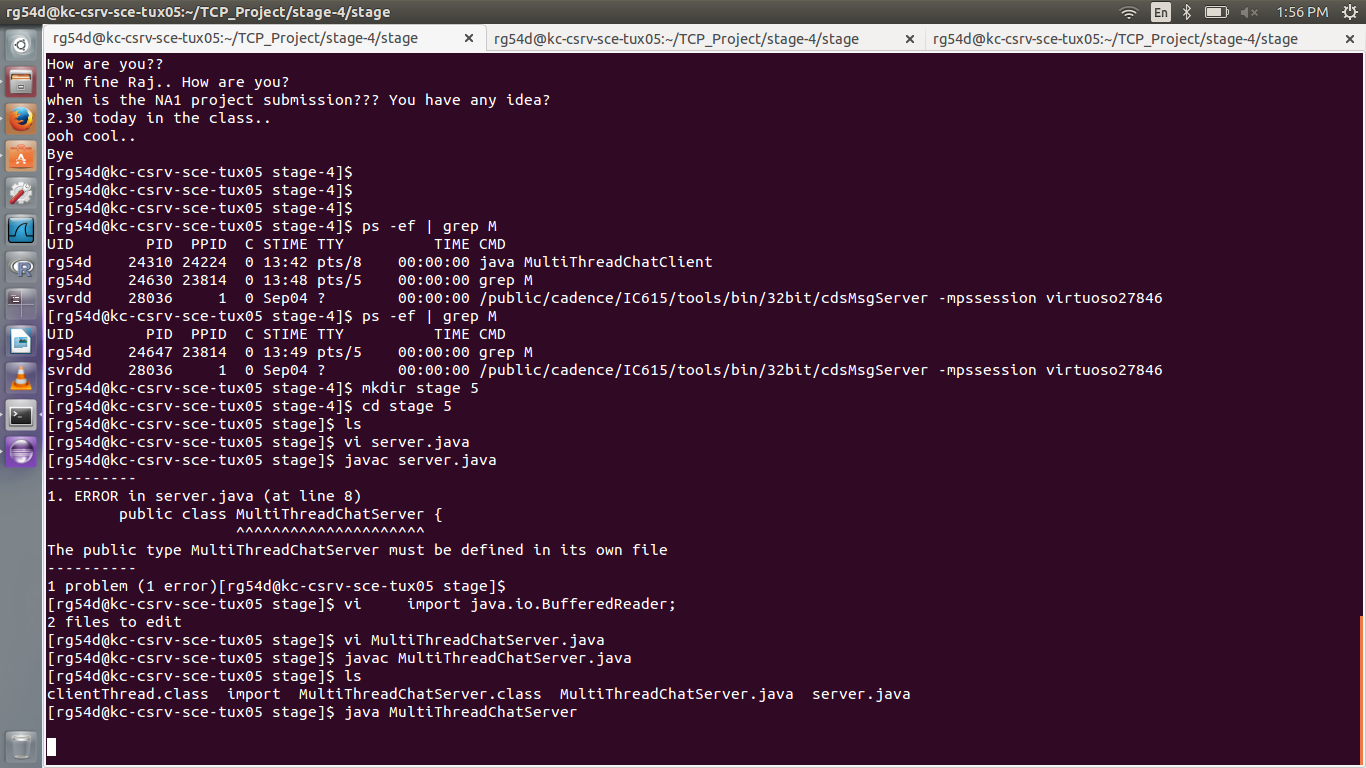
**d) The final one of phase two will be similar to previous one but here, servers will echo all msgs typed by all**

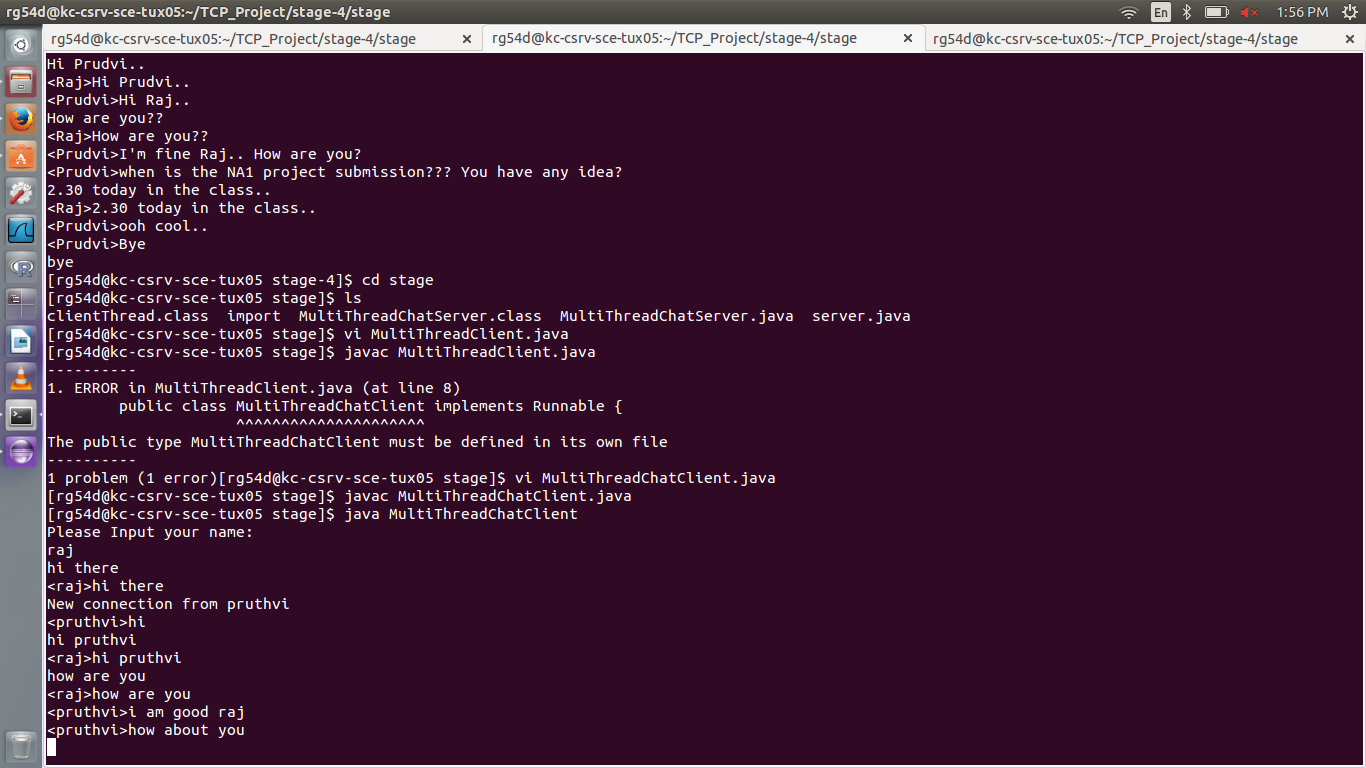
**Execution:**

* Server is running(listening)
* Message to all clients when opened(type your name)
* Client give name
* Echoed to all other nods
* Other client gives name
* Echoed to all clients
* Clients type msgs
* Sent to all clients connected to nodes

**Screen Shots**



™



**Tux login:**

**(An alternate to Geni for this project)**

**Login to tux.sce.umkc.edu:**

ID and Password are umkc SSID and password respectively

Tux has many ip’s for it

Initially for multiple clients I was logged with different ip address

**References**

* **http://www.oracle.com/technetwork/java/socket-140484.html**
* **http://stackoverflow.com/questions/7620833/server-client-program-in-java**
* **https://blackboard.umkc.edu/bbcswebdav/pid-1800328-dt-content-rid-7700015\_1/courses/UMKC-CSEE5110-0001-46445-FS2015/Socket%20Programming%20Fall%202015.pdf**
* **https://blackboard.umkc.edu/webapps/blackboard/content/listContent.jsp?course\_id=\_125267\_1&content\_id=\_1631276\_1&mode=reset**
* [**http://groups.geni.net/geni**](http://groups.geni.net/geni)