Cycle-1 : Shell Scripting Date : 27/01/2020

0:UNIXcommands

Problem 0: UNIX COMMANDS

AIM:

Try out the following unixcommands(use manual and help features for support).

1. echo, read
2. more, less
3. man
4. chmod, chown
5. cd, mkdir, pwd, ls, find
6. cat, mv, cp, rm
7. wc, cut, paste
8. head, tail, grep, expr
9. Redirections & Piping
10. useradd, usermod, userdel, passwd
11. tar

PROGRAM CODE:

echo read	Echo displays line of text Syntax : Echo [string] Read is used to reads the contents of a line into a variable. Syntax : Read text
more less	More is used to view the text file in command prompt displaying one screen at a time in case file is large. Syntax : more -d sample.txt Less is used to displays file contents or command output one page at a time in your terminal
man	man is used to display the user manual of any **command** that we can run on the terminal Syntax : man chown
Chmod	allows you to change the permissions on a file using either a symbolic or numeric mode or a reference file Syntax : chmod 766 ex.txt

chown	Chown allows you to change the user and/or group ownership of a given file, directory, or symbolic link Syntax : sudochown clave : mary example.txt
cd mkdir pwd ls find	cd is used to change the current working directory Syntax : cd directory_name Mkdir allows the user to create directories Syntax : mkdir invoice Pwd printing the current working directory Syntax :pwd Ls is used to listing the contents of a directory Syntax : ls–l Find is used used to search and locate the list of files and directories based on conditions you specify for files that match the arguments Syntax : find . name * ones
cat mv cp rm	allows us to create single or multiple files, view contain of file, concatenate files and redirect output in terminal or files Syntax : cat [OPTION]..[FILE] used to move one or more files or directories from one place to another in file system Syntax : mv[OPTION]source copying files and directories Syntax : cp [OPTION] source copying files and directories removing files and directories Syntax:rm [OPTION]…[FILE]
wc cut paste	wc is used for printing newline , word and byte counts for files. cut is used for cutting out sections for each line of files and writes result to standard output. Paste is used to join files horizontally by outputting lines.
head tail	outputting the first part of files given to it via standard input. displays the last part (10 lines by default) of one or more files or piped data

grep expr	grep searches through a set of files for arbitrary text pattern through regular expression. expr evaluates a given expression and displays corresponding output.
redirection piping	Redirection is a feature when executes a command,we can change input or output devices. The basic workflow of any Linux command is that it takes an input and give an output. • The standard input (stdin) device is the keyboard. • The standard output (stdout) device is the screen. With redirection, the above standard input/output can be changed. Pipe is used to combine two or more commands, and in this, the output of one command acts as input to another command, and this command's output may act as input to the next command and so on
Useradd usermod userdel Passwd	used for adding/creating user accounts in ***Linux*** and other Unix-like operating systems used to modify or change any attributes of a already created user account via ***command*** line used to delete a user account and related files used to change the user account passwords
Tar	Tar stands for tap to achieve which is used to tape drive back up command used by linux. Syntax : tar [OPTIONS] [ARCHIEVE-FILE] [FILE OD DIRECTORY TO BE ACHIEVED]

1:welcome message **Date :02/02/2020**

Problem 1 : WELCOME MESSAGE

AIM:

Print a customized welcome message. Get the name of the user as input and attach the name to the welcome message.Eg. “Welcome Rahul”.

PROGRAM CODE:

welcome	read -p “Enter your name “ name echo Welcome $name

OUTPUT:

Enter your name Preethi

Welcome Preethi

RESULT:

Through this script I can understand that how to take input and what is the syantax for that.

2:Greatest of2numbers **Date :06/02/2020**

Problem 2 : GREATEST OF 2 NUMBERS

AIM:

Take 2 numbers as input and print the greater of the two.

PROGRAM CODE:

Greater	read -p "Enter first number " num1 read –p "Enter second number " num2 if[$num1 –gt $num2] then echo " Largest number is :" $num1 else echo "Largest number is :" $num2 fi

OUTPUT:

Enter first number 100

Enter second number 45

Largest number is : 100

RESULT:

In this script I am inputting two numbers and find the largest number among those numbers.Through this program I can understand how to use conditional statements.

3:Oddnumbers **Date :09/01/2020**

Problem 3: ODD NUMBERS

AIM:

Print the first 20 odd numbers.

PROGRAM CODE:

Odd	i=1 count=0 echo "Odd numbers are :" while[$i –lt 100] do echo $i count=` expr$count+1 ` if[$count –eq 20] then break fi i=` expr $i+2' done

OUTPUT:

Odd numbers are :

1

3

....

39

RESULT:

Through this script I can understand how to use looping statements.

4: Sum of20numbers **Date :16/02/2020**

Problem 4 : SUM OF 20 NUMBERS

AIM:

Store 20 numbers in an array and print their sum.

PROGRAM CODE:

arraysum	read –p "Enter 20 numbers" input sum=0 for i in ${input[@]} do sum=` expr $sum+$i ` done echo "The sum is :" $sum

OUTPUT:

Enter 20 numbers 1 2 3 4 5 6 1 2 3 4 1 2 3 4 5 6 7 8 9 1

The sum is : 77

RESULT:

Through this script I can understand how to use array and the usage of looping statements.

5:Creating a textfile **Date :16/02/2020**

Problem 5 :CREATING A TEXT FILE

AIM:

Create a text file with 20 lines of text.

PROGRAM CODE:

text	cat >question5.txt Redirection is a feature in Linux such that when executing a command, you can change the standard input/output devices. The basic workflow of any Linux command is that it takes an input and give an output. With redirection, the above standard input/output can be changed.Pipe is used to combine two or more commands, and in this, the output of one command acts as input to another command, and this command's output may act as input to the next command and so on . Cntrl + D

OUTPUT:

File created que5.txt.

RESULT:

In this script I am creating file with name question5.txt,and in that file creating 20 lines of text.

6:Replacing strings **Date :20/02/2020**

Problem 6:REPLACING STRING

AIM:

Open the file created in question 5 and replace any string with another without using stream editor.

PROGRAM CODE:

replace	While read a; do echo ${a/command//output} done<question5.txt>question5.txt.t mv question5.txt{.t,}

OUTPUT:

Redirection is a feature in Linux such that when executing a output, you can change the standard input/output devices. The basic workflow of any Linux output is that it takes an input and give an output. With redirection, the above standard input/output can be changed.Pipe is used to combine two or more outputs, and in this, the output of one command acts as input to another output, and this output's output may act as input to the next output and so on

RESULT:

In this script I am replacing the word command with output.

7:Protocolsanddescription **Date :23/02/2020**

PROGRAM 7 : PROTOCOLS AND DESCRIPTION

AIM:

Open the /etc/protocols file and copy the protocol number of the following protocols into another file named "favorite protocols" and format it in the same way as the original /etc/protocol file.

1. udp
2. idrp
3. skip
4. ipip

PROGRAM CODE:

protocol	grep “udp|idrp|skip|ipip” /etc/protocols> favoriteprotocols.txt cat favoriteprotocols.txt

OUTPUT:

```
idrp    17      UDP     #User Datagram Protocol
idrp    45      IDRP    #Inter- Domain Routing Protocol
skip    57      SKIP    #SKIP
ipip    94      IPIP    #IP- within- IP Encapsulation Protocol
```

RESULT:

In this script I am opening the file /etc/protocols and copyies the protocol number of the udp,idrp,skip and ipip protocols to the another file “favorite protocols”.

8:Using ATandBATCH **Date :27/02/2020**

Problem 8:USING AT AND BATCH

AIM:

Use "at" and "batch" to schedule tasks.

PROGRAM CODE:

at	echo “sh execute.sh” |at now+1 minute
execute.sh	echo “ Hello world” >create.txt
Batch (In Terminal)	>batch Warning: commands will be executed using /bin/sh at> echo “ welcome…. >out.txt at>cntrl + D

OUTPUT:

job1 at Mon Feb 27 21:55:00 2020

The file named execute.sh will be executed and create.txt will be created after one minute.dd

job 2 at Mon Feb 27 22:05:00 2020

The file named out.txt will be created .

RESULT:

Using this script can understand how to use “at” and “batch”.

9:CRONcommand **Date :02/03/2020**

Problem 9: CRON COMMAND

AIM:

Use cron to schedule tasks.

PROGRAM CODE:

In terminal	Crontab –e
In nano editor	17 * * * * cd/home/mca47/Downloads/sd&&sh execute.sh
execute.sh	echo “Computer Programming“>computer.txt

OUTPUT:

When time (minutes) becomes 17 execute.sh will be execute. In execute.sh , we wants to create a file named as computer.txt with text “ Computer Programming“.

RESULT:

Through this scriptI can understand what is the use of cron command.

11:UNIXMail **Date :05/03/2020**

Problem 10: UNIX MAIL

AIM:

Set up unix mail and use mail to send and receive mails to and from users using shell scripting.

PROGRAM CODE:

Que22.sh	echo “hi preethiiii, welcome “|mail –s “subject ” preethips97@gmail.com

OUTPUT:

In gmail

19mca47

subject

Hi prwethiiii, welcome

RESULT:

Through this script I can understand how to setup unix mail and how to send and receive the mail also.

1:Gitrepository **Date :09/03/2020**

Cycle 2-Version Control using git

PROGRAM 1 : GIT REPOSITORY

AIM:

0. Install and initialize git and perform the followingoperations

a. Create a text file in your gitdirectory.

b. Configure your git with yourcredentials.

c. Configure the default editor to your favoriteeditor

d. Stage yourfiles

e. Create your firstcommit

f. Push to remoterepository

The following exercises must be done by a team of four students.

1. Create teamaccount.
2. Create empty repository in any git remote repository service andadd collaborators.
3. Leader must create the first commit.
4. All members must clone the remoterepository.
5. Each member must create a feature branch each and add features to them(anymod)
6. Commit changes tobranches.
7. Push thebranches.
8. View Graph.
9. Leader must make changes to themaster.
10. All member must rebase their branches to the position of latest commit inmaster.
11. Merge all branches tomaster.

12.Cherry pick commits from each branch createdearlier.

13.View Status.

14.View History.

15.Delete allbranches.

PROGRAM CODE :

Setting the git configurations in terminal.

- First configuring the username as :

  git config --global user.name

  kukkypriya

- second configuring the email id as :

  git config --global user.email

  preethips97@gmail.com

- setting nano as default editor as:

  git config --global core.editnano

- Adding a new file in our git folder ,Using the command :

  git add file1.txt

- Using the following command we can find the tracked and untracked files as:

  git status

- To commit the added file to local repository, Use command:

  git commit -m "a meaninful comment".

- To clone the entire githubrepository, use command:

  git clone https://github.com/Anjali-941/first-repo.git.

- Before pushing ,pull is performed for updating the git folder.

  git pull https://github.com/Anjali-941/first-repo.git

- To push the committed file to our gitrepository , use the below command:

  git push https://github.com/Anjali-941/first-repo.git master

- To create a new branch and branch operations. Use the following commands

  - git checkout -b pree /*to create a branch named revs*/
  - gitbranch /*to show allbranches*/
  - git pull https://github.com/Anjali-941/first-repo.git
  - git push https://github.com/Anjali-941/first-repo.git pree

- To merge the branch to our master branch,use following commands.

  git checkout master /*switching to master branch*/

  git merge pree /*merging the branchrepos*/

- To cherry pick a commit done in revs branch to master branch, use the command :

  git cherry-pick commitid

- To view the operations in a graph format , use the command:

  git log –-graph

- To rebase the branch to master branch , use the command :

  git checkout pree

  git rebase master /* rebasing branch to master branch*/

- To view the history of git commits, use the command :

  git log --oneline

- To delete a feature branch , use the command :

  git branch -d pree /*deletes repos branch*/

RESULT :

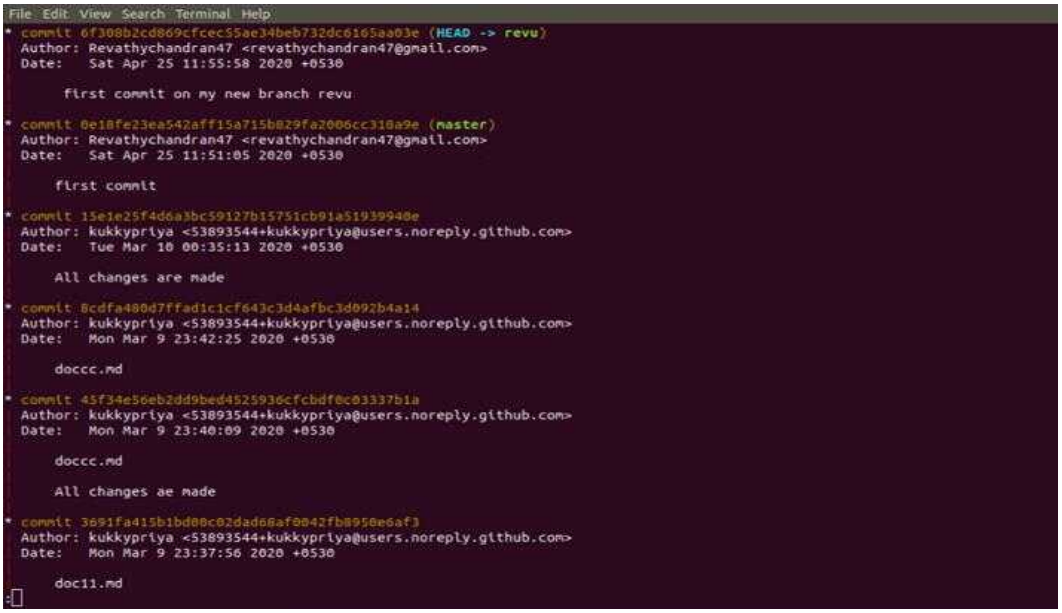

Figure 1 :Git graph,Output of git status command

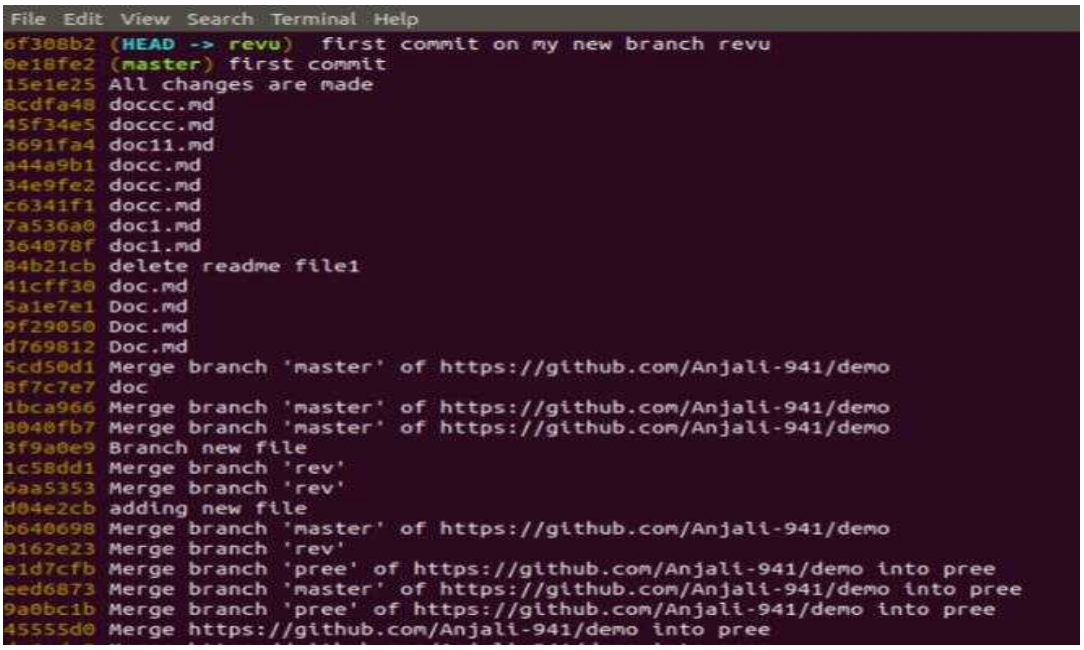

Figure 2 :Git status

Cycle-3-Network Programming In Java

1:TCPclient-server Date :12/3/2020

Problem 1:TCP CLIENT - SERVER

AIM:

Implement Bidirectional Client-Server communication using TCP.

PROGRAM CODE:

Server2.java	import java.io.*;
	import java.net.*;
	class Server2 {
	
	public static void main(String args[])
	throws Exception
	{
	ServerSocketss = new ServerSocket(888);
	Socket s = ss.accept();
	System.out.println("Connection
	established");
	
	PrintStreamps = new
	PrintStream(s.getOutputStream());
	
	BufferedReaderbr = new
	BufferedReader(newInputStreamReader
	(s.getInputStream()));
	
	BufferedReader kb = new
	BufferedReader(new InputStreamReader
	(System.in));

	while (true) {
	String str, str1;
	while ((str = br.readLine()) != null)
	{
	System.out.println(str);
	str1 = kb.readLine();
	
	ps.println(str1);
	}
	
	ps.close();
	br.close();
	kb.close();
	ss.close();
	s.close();
	System.exit(0);
	
	}
	}
	}

Client2.java	import java.io.*;
	import java.net.*;
	
	class Client2 {
	
	public static void main(String args[])
	throws Exception
	{
	
	Socket s = new Socket("localhost", 888);
	DataOutputStream dos = new
	DataOutputStream(s.getOutputStream());
	
	BufferedReaderbr = new
	BufferedReader(new InputStreamReader(
	s.getInputStream()));

```
                    BufferedReader kb = new
            BufferedReader(new InputStreamReader
                    (System.in));

                    String str, str1;
                    while (!(str = kb.readLine()).
equals("exit")) {

                            dos.writeBytes(str + "\n");

                            str1 = br.readLine();

                            System.out.println(str1);
                    }


                    dos.close();
                    br.close();
                    kb.close();
                    s.close();
            }
}
```

RESULT:

We are creating a local client and server communication. First running server program , if it is free of bugs it will wait for a client to connect. Then we run client program , if it is correct then a connection between client and server will be established.

SCREENSHOTS:

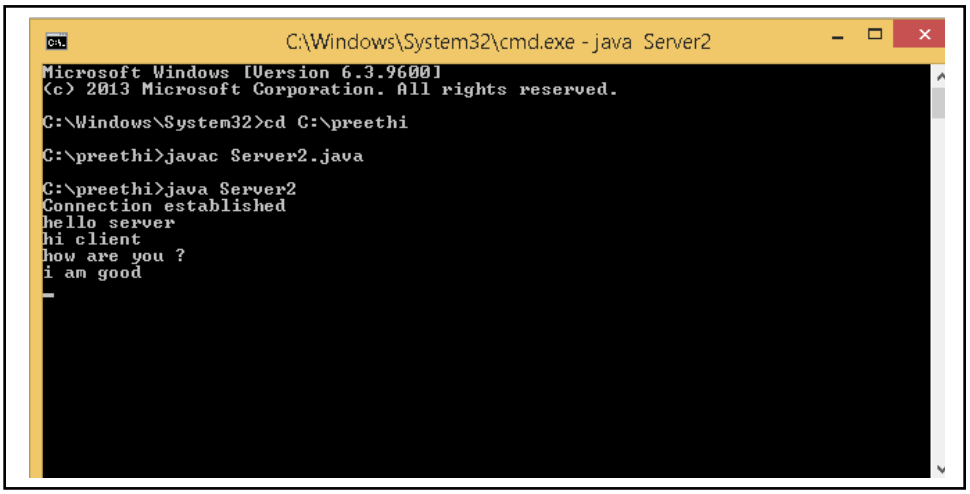

Figure 1: This is the local server which communicates with client machine after the establishment of connection between client and server.

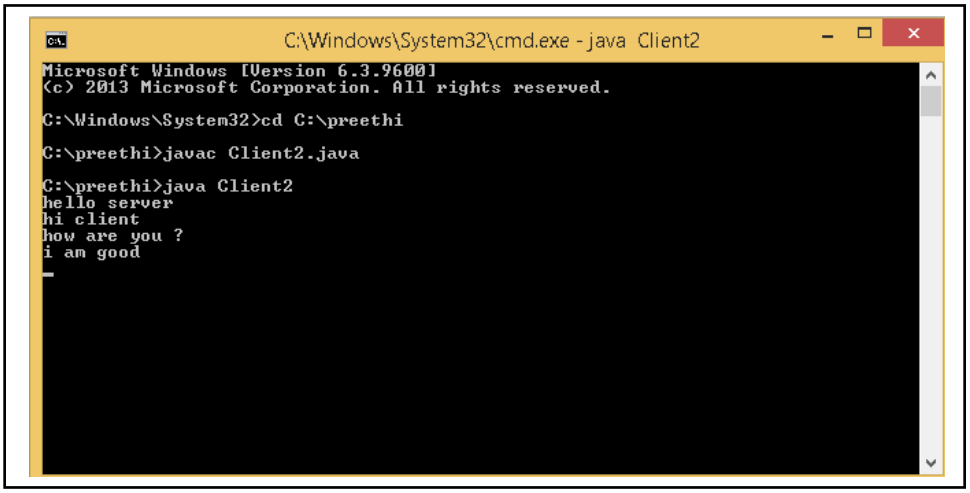

Figure 2: This is the local client which communicates with server machine after the establishment of connection between client and server.

2:TCPechoserver **Date:16/3/2020**

Problem 2:TCP ECHO SERVER

AIM:

Implement Echo Server using TCP.

PROGRAM CODE:

EchoServer.java	import java.io.*;
	import java.net.*;
	public class EchoServer
	{
	public static void main(String args[]) throws
	Exception
	{
	try
	{
	int Port;
	BufferedReaderBuf =new BufferedReader(
	newInputStreamReader(System.in));
	
	System.out.print(" Enter the Port
	Address : ");
	
	Port=Integer.parseInt(Buf.readLine());
	ServerSocketsok =new ServerSocket
	(Port);
	
	System.out.println(" Server is Ready To
	Receive a Message. ");
	System.out.println(" Waiting ..... ");
	Socket so=sok.accept();

	if(so.isConnected()==true)
	System.out.println(" Client Socket is
	Connected Succecfully. ");
	
	InputStream in=so.getInputStream();
	OutputStreamou=so.getOutputStream();
	PrintWriterpr=new PrintWriter(ou);
	BufferedReaderbuf=newBufferedReader
	
	(newInputStreamReader(in));
	
	String str=buf.readLine();
	System.out.println(" Message Received
	From Client : " + str);
	
	System.out.println(" This Message is
	Forwarded To Client. ");
	
	pr.println(str);
	pr.flush();
	}
	catch(Exception e)
	{
	System.out.println(" Error : " +
	e.getMessage());
	}
	}
	}

EchoClient.java	import java.io.*; import java.net.*; public class EchoClient { public static void main(String args[]) throws

```
    Exception
{
try
{

int Port;
BufferedReaderBuf =newBufferedReader
        (newInputStreamReader(System.in));

System.out.print(" Enter the Port
        Address : " );

Port=Integer.parseInt(Buf.readLine());
Socket sok=new Socket("localhost",
        Port);



if(sok.isConnected()==true)
System.out.println(" Server Socket is
        Connected Succecfully. ");

InputStream in=sok.getInputStream();
OutputStreamou=sok.getOutputStream();

PrintWriterpr=new PrintWriter(ou);

BufferedReader buf1=newBufferedReader
(newInputStreamReader (System.in));
BufferedReader buf2=new
BufferedReader(newInputStreamReader
        (in));

String str1,str2;
System.out.println(" Enter the Message :
        ");

str1=buf1.readLine();

pr.println(str1);
pr.flush();

System.out.println(" Message Send
        Successfully. ");
```

```
str2=buf2.readLine();

System.out.println(" Message From
        Server : " + str2);
}
catch(Exception e)
{
System.out.println(" Error : " +
e.getMessage());
 }
}
```

RESULT:

We are creating a local client and echo server communication. First running echo server program , if it is free of bugs it will wait for a client to connect. Then we run client program , if it is correct then a connection between client and echo server will be established. In our program , the message received from client is forwarded to client itself by the echo server.

SCREENSHOTS:

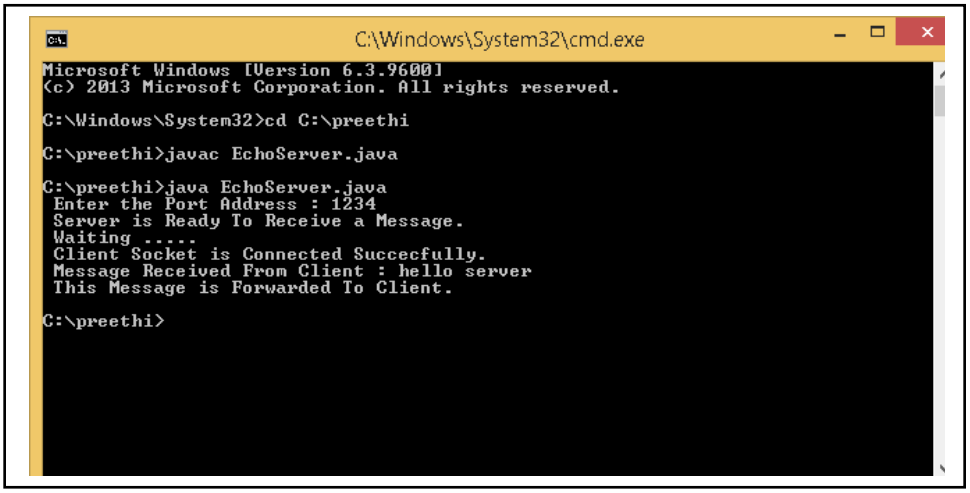

Figure 1: This is the echo server which communicates with Client machine after the establishment of connection between client and server. The message received is forwarded to client.

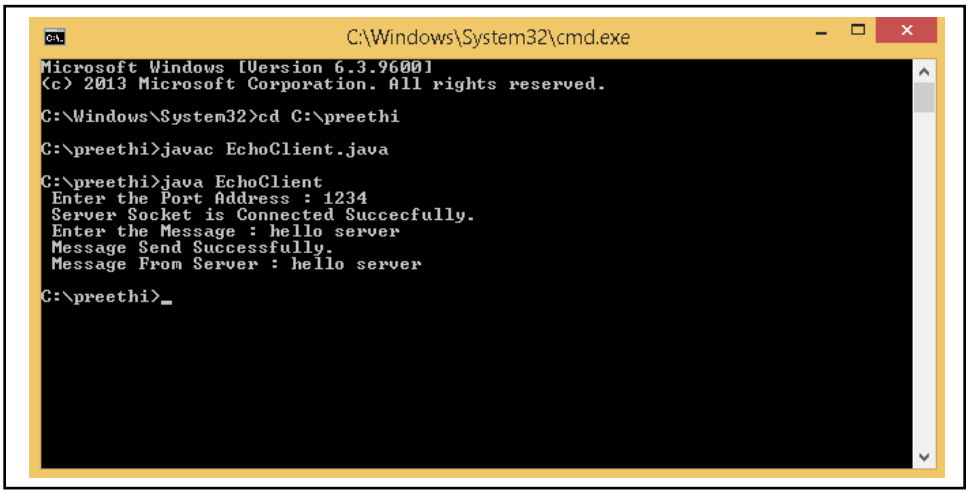

Figure 2: This is the client which communicates with echo server after the establishment of connection between client and server.

3:UDPChatserver **Date :19/03/2020**

Problem 3:CHAT SERVER USING UDP

AIM:

Implement Chat Server using UDP.

PROGRAM CODE:

UDPServer.java	import java.io.*;
	import java.net.*;
	classUDPServer
	{
	public static DatagramSocketserversocket;
	public static DatagramPacketdp;
	public static BufferedReader dis;
	public static InetAddressia;
	public static byte buf[] = new byte[1024];
	public static intcport = 789,sport=790;
	public static void main(String[] a) throws
	IOException
	{
	serversocket = new DatagramSocket(sport);
	dp = new DatagramPacket(buf,buf.length);
	dis = new BufferedReader
	(newInputStreamReader(System.in));
	ia = InetAddress.getLocalHost();
	System.out.println("Server is Running...");
	while(true)
	{
	serversocket.receive(dp);
	String str = new String(dp.getData(), 0,

	dp.getLength());
	if(str.equals("STOP"))
	{
	System.out.println("Terminated...");
	break;
	}
	System.out.println("Client: " + str);
	String str1 = new String(dis.readLine());
	buf = str1.getBytes();
	serversocket.send(new
	DatagramPacket(buf,str1.length(), ia,
	cport));
	}
	}
	}

UDPClient.java	import java.io.*;
	import java.net.*;
	
	class UDPClient
	{
	public static DatagramSocketclientsocket;
	public static DatagramPacketdp;
	public static BufferedReader dis;
	public static InetAddressia;
	public static byte buf[] = new byte[1024];
	public static intcport = 789, sport = 790;
	public static void main(String[] a) throws
	IOException
	{
	clientsocket = new DatagramSocket(cport);
	dp = new DatagramPacket(buf, buf.length);
	
	dis = new BufferedReader(new
	InputStreamReader(System.in));
	ia = InetAddress.getLocalHost();
	
	System.out.println("Client is Running... Type
	'STOP'to Quit");
	
	while(true)

```
{
String str = new String(dis.readLine());
buf = str.getBytes();
if(str.equals("STOP"))
{
System.out.println("Terminated...");



clientsocket.send(new
DatagramPacket(buf,str.length(), ia,
sport));
break;
}

clientsocket.send(new DatagramPacket(buf,
str.length(), ia, sport));

clientsocket.receive(dp);

String str2 = new String(dp.getData(), 0,
dp.getLength());

System.out.println("Server: " + str2);
}

}

}
```

RESULT:

We are creating a local client and echo server communication using UDP. First running server program , if it is free of bugs it will wait for a client to connect. Then we run client program , if it is correct then a connection between client and server will be established. Thus after the successful establishment of connection between them , they can send and receive messages as in a chat.

SCREENSHOTS:

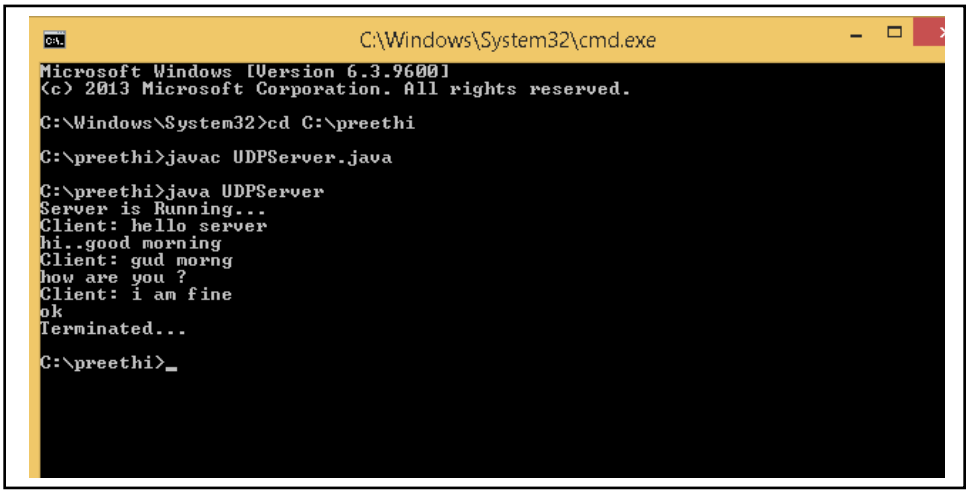

Figure 1:The figure shows the local server. Server will wait for the client to connect. After the client has connected , client can send a message to server. It will be shown as a chat .

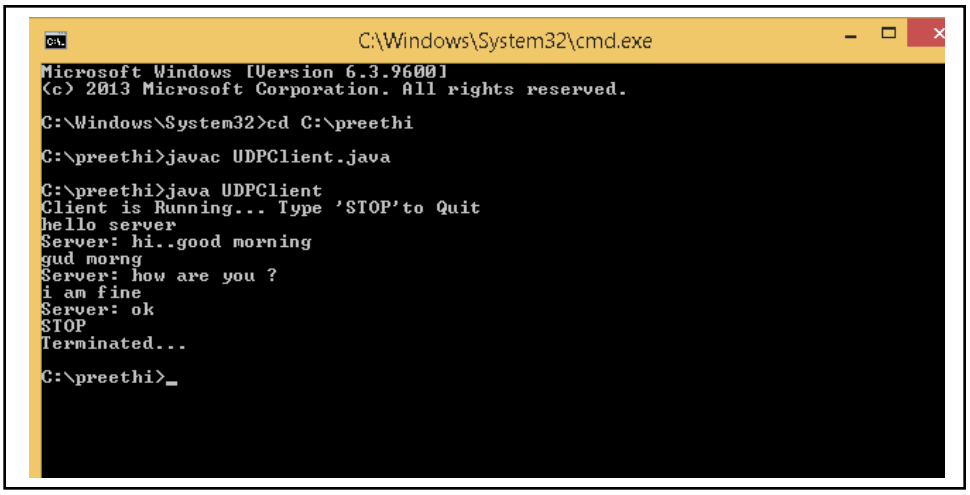

Figure 2: The figure shows the local client. After the client has connected to the server , client can send a messageto server. It will be shown as a chat . The chat can be stopped if client can send a message to server as STOP.