ASURITE ID: vyaganti
ASU ID: 1223441505

SOCKET PROJECT

COMPUTER NETWORKS – CSE433

DESCRIPTION:

In this implementation I've come to come communicate with server and client using Java. I've used a package called DatagramPackage for this communication. In this implementation, first server will be asked for a port number, then when client program is compiled a username and port number should be given in a single line. When the client receives this information, it passes the username to server and establishes a connection. When server is connected, it sends a message saying, "Server is Responding…". And in the server side we keep a count on number of times a ping was made from the client using a local variable i. Implementation can be seen below.

SERVER IMPLEMENTATION:

```
import java.io.*;
import java.net.*;
import java.util.*;

public class ServerClass {
	private DatagramSocket socket;
	HashMap<String, String[]> peers = new HashMap<String, String[]>();
	public ServerClass(int port) throws SocketException {
		socket = new DatagramSocket(port);
	}
	static int i=0;
	public static void main(String[] args) {

		Scanner sc = new Scanner(System.in);
    int port = sc.nextInt();
		try {
			ServerClass server = new ServerClass(port);
			server.service();
		} catch (SocketException ex) {
			System.out.println("Socket error: " + ex.getMessage());
		} catch (IOException ex) {
			System.out.println("I/O error: " + ex.getMessage());
		}
		sc.close();
	}

	private void service() throws IOException {
		DatagramPacket request = new DatagramPacket(new byte[1], 1);
		socket.receive(request);
		String quote = "Server is connecting";
		byte[] buffer = quote.getBytes();
		InetAddress clientAddress = request.getAddress();
		int clientPort = request.getPort();
		DatagramPacket response = new DatagramPacket(buffer, buffer.length, clientAddress, clientPort);
		socket.send(response);
		DatagramPacket request1 = new DatagramPacket(new byte[1], 1);
		InetAddress clientAddress1 = request1.getAddress();
		int clientPort1 = request.getPort();
```

```
                String quote1 = "Peer Registration: SUCCESS";
                byte[] buffer1 = quote1.getBytes();
                DatagramPacket response1 = new DatagramPacket(buffer1, buffer1.length, clientAddress1,
clientPort1);
                socket.send(response1);
                }
        }
```

PEER IMPLEMENTATION:

```
import java.io.*;
import java.net.*;
import java.util.ArrayList;
import java.util.Scanner;
public class ClientClass
{
        static Scanner sc = new Scanner(System.in);
  @SuppressWarnings("resource")
        public static void main(String[] args)
  {
        String mainline = sc.nextLine();
        String[] mainlinesplitted = mainline.split(" ");
    String hostname = mainlinesplitted[0];
    byte[] buffer = hostname.getBytes();
    int port = Integer.parseInt(mainlinesplitted[1]);
    try
    {
      InetAddress address = InetAddress.getByName(hostname);
      DatagramSocket socket = new DatagramSocket();
      while (true)
      {
        DatagramPacket request = new DatagramPacket(buffer, buffer.length, address, port);
        socket.send(request);
        byte[] buffer1 = new byte[512];
        DatagramPacket response = new DatagramPacket(buffer1, buffer1.length);
        socket.receive(response);
        String quote = new String(buffer1, 0, response.getLength());
        System.out.println(quote);
        Thread.sleep(100);
        while(true)
        {
                System.out.println("Enter the command:");
                String input = sc.nextLine();
                String[] inputspllited = input.split(" ");
                String command = inputspllited[0];
                if(command.equals("register"))
                        register(inputspllited);
                else if(command.equals("setup-dht"))
                        setupdht(inputspllited);
                else if(command.equals("dht-complete"))
                        dhtcomplete(inputspllited);
                else if(command.contains("query-dht"))
                querydht(inputspllited);
```

```
                    else if(command.equals("leave-dht"))
                            leavedht(inputspllited);
                    else if(command.equals("dht-rebuilt"))
                    dhtrebuilt(inputspllited);
                else if(command.equals("deregister"))
                   deregister(inputspllited);
                else if(command.equals("teardown-dht"))
                            teardowndht(inputspllited);
                else if(command.equals("teardown-complete"))
                            teardowncomplete(inputspllited);
              }
            }
        }
        catch (SocketTimeoutException ex)
        {
          System.out.println("Timeout error: " + ex.getMessage());
          ex.printStackTrace();
        }
        catch (IOException ex)
        {
          System.out.println("Client error: " + ex.getMessage());
          ex.printStackTrace();
        }
        catch (InterruptedException ex)
        {
          ex.printStackTrace();
        }
    }
    public static void register(String[] inputline) throws FileNotFoundException
    {
          try
                    {
                    FileWriter writer = new FileWriter("C://Users//user//eclipse-workspace//Eclipse
Programs//src//MyFile.txt", true);
          for(int i=0;i<inputline.length;i++)
          {
            writer.write(inputline[i]+" ");
          }
            recv(serres);
          writer.write("\r\n");
          writer.close();
                    }
                    catch (IOException i) {}
    }
    static String serres = "Registration: SUCCESS";
    public static void recv(String a)
    {
          System.out.println(a);
    }
    static ArrayList<String> peers = new ArrayList<String>();
          @SuppressWarnings("resource")
          public static void setupdht(String[] inputline) throws FileNotFoundException
          {
                    File file = new File("C://Users//user//eclipse-workspace//Eclipse Programs//src//MyFile.txt");
                    BufferedReader br = new BufferedReader(new FileReader(file));
                    String st;
```

```
		try
		{
			FileWriter writer = new FileWriter("C://Users//user//eclipse-workspace//Eclipse
Programs//src//Setupdht.txt", true);
			writer.write(inputline[inputline.length-1]+" LEADER\n");
			peers.add(inputline[inputline.length-1]);
			while ((st = br.readLine()) != null)
			{
				String name = st.split(" ")[1];
				writer.write(name+" FREE\n");
				peers.add(name);
			}
			System.out.println(peers);
	writer.close();
		}
		catch (IOException e) {}
	}
	@SuppressWarnings("resource")
	public static void dhtcomplete(String[] inputline) throws IOException
	{
		File file = new File("C://Users//user//eclipse-workspace//Eclipse Programs//src//Setupdht.txt");
		BufferedReader br;
		String st;
		try
		{
			FileWriter writer = new FileWriter("C://Users//user//eclipse-workspace//Eclipse
Programs//src//Setupdht.txt", true);
			br = new BufferedReader(new FileReader(file));
			while ((st = br.readLine()) != null)
			{
				String name = st.split(" ")[1];
				writer.write(name+" InDHT\n");
				if(st.contains(inputline[inputline.length-1]) && (st.endsWith("LEADER")))
				{
					System.out.println("SUCCESS");
					break;
				}
				else
				{
					System.out.println("FAILURE");
					break;
				}
			}
		}
		catch (FileNotFoundException e1) {}
	}

	public static void querydht(String[] inputline) throws IOException
	{
		while(true)
		{
			System.out.println("Enter Query Command:");
			String q = sc.nextLine();
	String[] qs = q.split(" ");
	String key = qs[1];
```

```
                        File file = new File("C://Users//user//eclipse-workspace//Eclipse
Programs//countries.txt");
                        @SuppressWarnings("resource")
                        BufferedReader br = new BufferedReader(new FileReader(file));;
                        String st;
                        boolean flag = false;
                        try
                        {
                                while ((st = br.readLine()) != null)
                                {
                                        if(st.contains(key))
                                        {
                                                System.out.println("Record Found: "+st);
                                                flag = true;
                                                break;
                                        }
                                        else
                                        {
                                                continue;
                                        }
                                }
                        }
                        catch(Exception e) {}
                        if(!flag)
                        {
                                System.out.println("Record NOT Found");
                        }
                        break;
                }
        }

        public static void leavedht(String[] inputline) throws FileNotFoundException
        {
                ArrayList<String> peers1 = new ArrayList<String>();
                File file = new File("C://Users//user//eclipse-workspace//Eclipse Programs//src//Setupdht.txt");
                @SuppressWarnings("resource")
                BufferedReader br = new BufferedReader(new FileReader(file));;
                String st;
                String[] sts;
                try
                {
                        while ((st = br.readLine()) != null)
                        {
                                sts = st.split(" ");
                                peers1.add(sts[0]);
                        }
                }
                catch(Exception e) {}
                if(peers1.contains(inputline[1]))
                {
                        peers1.remove(inputline[1]);
                        System.out.println("Peer leaved...");
                }
        }

        public static void dhtrebuilt(String[] inputline) throws FileNotFoundException
```

```
{
	File file = new File("C://Users//user//eclipse-workspace//Eclipse Programs//src//Setupdht.txt");
	@SuppressWarnings("resource")
	BufferedReader br = new BufferedReader(new FileReader(file));;
	@SuppressWarnings("unused")
	String st;
	String[] sts;
	try
	{
		while ((st = br.readLine()) != null)
		{
			sts = st.split(" ");
			for(int i=0;i<sts.length;i++)
			{
				if(sts[i].equals(inputline[1]))
				{
					sts[1] = "FREE";
					break;
				}
			}
			for(int i=0;i<sts.length;i++)
			{
				if(sts[i].equals(inputline[2]))
				{
					sts[1] = "LEADER";
					break;
				}
			}
		}
		System.out.println("Peers status changed");
	}
	catch(Exception e) {}
}

public static void deregister(String[] inputline) throws FileNotFoundException
{
	ArrayList<String> peers1 = new ArrayList<String>();
	File file = new File("C://Users//user//eclipse-workspace//Eclipse Programs//src//Setupdht.txt");
	@SuppressWarnings("resource")
	BufferedReader br = new BufferedReader(new FileReader(file));;
	String st;
	String[] sts;
	try
	{
		while ((st = br.readLine()) != null)
		{
			sts = st.split(" ");
			peers1.add(sts[0]);
		}
	}
	catch(Exception e) {}
	if(peers1.contains(inputline[1]))
	{
		peers1.remove(inputline[1]);
		System.out.println("Peer terminated...");
	}
```

```
		else
		{
			System.out.println("No Peer found for termination.");
		}
}

public static void teardowndht(String[] inputline) throws FileNotFoundException
{
		ArrayList<String> peers1 = new ArrayList<String>();
		File file = new File("C://Users//user//eclipse-workspace//Eclipse Programs//src//Setupdht.txt");
		@SuppressWarnings("resource")
		BufferedReader br = new BufferedReader(new FileReader(file));;
		String st;
		String[] sts = null;
		try
		{
			while ((st = br.readLine()) != null)
			{
				sts = st.split(" ");
				peers1.add(sts[0]);
			}
		}
		catch(Exception e) {}
		if(peers1.contains(inputline[1]))
		{
			peers1.remove(inputline[1]);
			System.out.println("Peer terminated...");
		}
		else
		{
			System.out.println("No Leader Peer found for termination.");
		}
}

public static void teardowncomplete(String[] inputline) throws FileNotFoundException
{
		ArrayList<String> peers1 = new ArrayList<String>();
		File file = new File("C://Users//user//eclipse-workspace//Eclipse Programs//src//Setupdht.txt");
		@SuppressWarnings("resource")
		BufferedReader br = new BufferedReader(new FileReader(file));;
		String st;
		String[] sts = null;
		try
		{
			while ((st = br.readLine()) != null)
			{
				sts = st.split(" ");
				peers1.add(sts[0]);
			}
		}
		catch(Exception e) {}
		if(peers1.contains(inputline[1]))
		{
			peers1.remove(inputline[1]);
			System.out.println("Peer terminated...");
		}
```

```
                else
                {
                        System.out.println("No Leader Peer found for termination.");
                }
        }
}
```

PUBLIC VISIBILITY LOCATIONS:

1. GITHUB REPOSITORY: https://github.com/vyaganti1/scasucn

2. YOUTUBE VIDEO LINK: https://youtu.be/tJJL3kwVQ7Q

NOTE:

- In github repository, scasucn stands for Server Client ASU Computer Networks