**CS550 Advanced Operating Systems****Programming Assignment 3**

**Source Code**

Submitted by:

Chiranjeevi Ankamredy

A20359837

# Indexingserver.java

**import java.io.\*;**

**import java.net.\*;**

**import java.io.BufferedReader;**

**import java.io.BufferedWriter;**

**import java.io.DataInputStream;**

**import java.io.InputStreamReader;**

**import java.util.Scanner;**

**import java.util.Hashtable;**

**import java.util.ArrayList;**

**import java.util.HashMap;**

**import java.util.Iterator;**

**import java.util.Map;**

**import java.util.Map.Entry;**

**import java.util.Set;**

**class Hash**

**{**

**public static int currentSize, maxSize;**

**public static String keys;**

**public static String vals;**

**public static Hashtable<String,String> data= new Hashtable<String,String>(1000001);**

**public Hash()**

**{ currentSize = 0;**

**keys = new String();**

**vals = new String();**

**}**

**void insert(String key, String val)**

**{**

**keys=key;**

**vals=val;**

**data.put(keys,vals);**

**return;**

**}**

**public String get(String Name){**

**keys=Name;**

**return data.get(keys);**

**}**

**void delete(String key)**

**{**

**keys=key;**

**data.remove(keys);**

**}**

**void printHashTable()**

**{ System.out.println("Hash Table " );**

**Iterator<Entry<String, String>> it = data.entrySet().iterator();**

**while (it.hasNext())**

**{**

**Entry<String, String> pair = it.next();**

**System.out.println(pair.getKey() + " " + pair.getValue());**

**}**

**}**

**}**

**class ThreadHandler extends Thread**

**{**

**Socket News;**

**int n;**

**ThreadHandler(Socket s,int v)**

**{**

**News=s;**

**n=v;**

**}**

**public void run()**

**{**

**try**

**{**

**System.out.println("Thread created for peer" );**

**Scanner scan = new Scanner(System.in);**

**DataInputStream inp = new DataInputStream(News.getInputStream());**

**DataOutputStream oup = new DataOutputStream(News.getOutputStream());**

**Hash h1 = new Hash();**

**char ch;**

**do**

**{**

**String ip = inp.readLine();**

**int choice2 = Integer.parseInt(ip);**

**switch (choice2)**

**{**

**case 1 :**

**// String ip31 = inp.readLine();**

**// int q = Integer.parseInt(ip31);**

**h1.insert(inp.readLine(), inp.readLine() );**

**String ip15="Success";**

**oup.writeBytes(ip15);**

**oup.writeByte('\n');**

**System.out.println("Files are inserted");**

**break;**

**case 2 :**

**String ip3 = inp.readLine();**

**String ip11=(String) h1.get(ip3);**

**if(ip11==null)**

**{ String i26="FileNotFound";**

**oup.writeBytes(i26);**

**oup.writeByte('\n');**

**}**

**else**

**{oup.writeBytes(ip11);**

**oup.writeByte('\n');**

**}**

**BufferedReader inFromClient = new BufferedReader(new InputStreamReader(News.getInputStream()));**

**DataOutputStream outToClient = new DataOutputStream( News.getOutputStream());**

**String Dfile = inp.readLine();**

**File myFile = new File (Dfile);**

**byte [] buffer = new byte [(int)myFile.length()];**

**FileInputStream fis = new FileInputStream(myFile);**

**BufferedInputStream bis = new BufferedInputStream(fis);**

**bis.read(buffer,0,buffer.length);**

**OutputStream os = News.getOutputStream();**

**System.out.println("Sending...");**

**os.write(buffer,0,buffer.length);**

**os.flush();**

**break;**

**default :**

**System.out.println("Wrong Entry ");**

**break;**

**}**

**h1.printHashTable();**

**String ctr = inp.readLine();**

**ch = ctr.charAt(0);**

**} while (ch == 'Y'|| ch == 'y');**

**// News.close();**

**}**

**catch(Exception e)**

**{**

**System.out.println(e);**

**}**

**}**

**}**

**public class IndexingServer**

**{**

**public static void main(String[] args)**

**{**

**int req=1001;**

**try**

**{**

**System.out.println("Enter The port of tthe server:");**

**Scanner x=new Scanner(System.in);**

**String port1=x.nextLine();**

**int port = Integer.parseInt(port1);**

**ServerSocket ss=new ServerSocket(port);**

**for(;;)**

**{**

**Socket s=ss.accept(); //establishes connection**

**System.out.println("Server started ");**

**Thread T =new ThreadHandler(s,req);**

**T.start();**

**req++;**

**}**

**}**

**catch(Exception e)**

**{System.out.println(e);}**

**}**

**}**

1. **peerserver.java**

**import java.io.\*;**

**import java.net.\*;**

**import java.util.Scanner;**

**class ThreadHandler extends Thread**

**{**

**Socket News1;**

**int n;**

**ThreadHandler(Socket s,int v)**

**{**

**News1=s;**

**n=v;**

**}**

**public void run()**

**{**

**try**

**{**

**DataInputStream inp1 = new DataInputStream(News1.getInputStream());**

**DataOutputStream oup1 = new DataOutputStream(News1.getOutputStream());**

**String fp = inp1.readLine();**

**if(fp.equals("upload"))**

**{**

**int filesize=266392;**

**int bytesRead;**

**int current = 0;**

**BufferedReader inFromUser = new BufferedReader(new InputStreamReader( System.in));**

**System.out.println("connected");**

**byte [] buffer = new byte [filesize];**

**InputStream is = News1.getInputStream();**

**FileOutputStream fos = new FileOutputStream("jay123.txt");**

**BufferedOutputStream bos = new BufferedOutputStream(fos);**

**// current = is.read(buffer,0,buffer.length);**

**//current = bytesRead;**

**// System.out.println(current);**

**do**

**{ bytesRead =is.read(buffer);**

**// current, (buffer.length-current));**

**bos.write(buffer, 0 , bytesRead);**

**System.out.println("file downloaded");**

**bos.flush();**

**//if(bytesRead >= 0)**

**//current += bytesRead;**

**System.out.println("-------------------");**

**} while(bytesRead >=0);**

**//bos.write(buffer, 0 , current);**

**//System.out.println("file downloaded");**

**//bos.flush();**

**bos.close();**

**}**

**if(fp.equals("download"))**

**{**

**BufferedReader inFromClient = new BufferedReader(new InputStreamReader(News1.getInputStream()));**

**DataOutputStream outToClient = new DataOutputStream( News1.getOutputStream());**

**String Dfile = inp1.readLine();**

**File myFile = new File (Dfile);**

**byte [] buffer = new byte [(int)myFile.length()];**

**FileInputStream fis = new FileInputStream(myFile);**

**BufferedInputStream bis = new BufferedInputStream(fis);**

**bis.read(buffer,0,buffer.length);**

**OutputStream os = News1.getOutputStream();**

**System.out.println("Sending...");**

**os.write(buffer,0,buffer.length);**

**os.flush();**

**}**

**}**

**catch(Exception e)**

**{**

**System.out.println(e);**

**}**

**}**

**}**

**class PeerServer {**

**public static void main(String args[]) throws Exception {**

**int req=101;**

**try**

**{**

**System.out.println("Enter The port of tthe server:");**

**Scanner x=new Scanner(System.in);**

**String port2=x.nextLine();**

**int port5 = Integer.parseInt(port2);**

**for(;;)**

**{**

**ServerSocket welcomeSocket = new ServerSocket(port5);**

**Socket connectionSocket = welcomeSocket.accept();**

**System.out.println("I m the Client server:");**

**Thread T =new ThreadHandler(connectionSocket,req);**

**T.start();**

**req++;**

**}**

**}**

**catch(Exception e)**

**{System.out.println(e);}**

**}**

**}**

1. **PeerClient.java**

**import java.io.\*;**

**import java.net.\*;**

**import java.io.BufferedReader;**

**import java.io.BufferedWriter;**

**import java.io.DataInputStream;**

**import java.io.File;**

**import java.io.FileInputStream;**

**import java.io.FileWriter;**

**import java.io.InputStreamReader;**

**import java.util.Scanner;**

**class Hashtableop**

**{**

**private int maxSize;**

**private String[] keys;**

**public Hashtableop(int capacity)**

**{**

**maxSize = capacity;**

**keys = new String[maxSize];**

**}**

**private int hash(String key)**

**{**

**return key.hashCode() % maxSize;**

**}**

**public int find(String key)**

**{**

**int tmp = hash(key);**

**return tmp;**

**}**

**}**

**class PeerClient {**

**public static void main(String args[])**

**{**

**try**

**{**

**Scanner scan = new Scanner(System.in);**

**Hashtableop Serverslct = new Hashtableop(8);**

**System.out.println("Enter key to connect server:");**

**int n=Serverslct.find(scan.next());**

**if (n < 0)**

**n = -n;**

**System.out.println("connecting to the server "+n);**

**int k=0;**

**BufferedReader br = new BufferedReader(new InputStreamReader(new FileInputStream("Config.txt")));**

**String Peerdetls;**

**String line;**

**while ((line = br.readLine()) != null)**

**{**

**if(n==k)**

**{**

**Peerdetls=line;**

**String words[] = Peerdetls.split(" ");**

**// String firstTwo = words[0] + " " + words[1];**

**System.out.println(words[0]);**

**System.out.println(words[1]);**

**int port = Integer.parseInt(words[1]);**

**Socket s=new Socket(words[0],port);**

**System.out.println("Peer1 Intitialized");**

**DataInputStream inp = new DataInputStream(s.getInputStream());**

**DataOutputStream oup = new DataOutputStream(s.getOutputStream());**

**char ch;**

**do**

**{**

**System.out.println("\nHash Table Operations\n");**

**System.out.println("1. Register ");**

**System.out.println("2. Search & Download");**

**System.out.println("Enter the Choice:");**

**int choice = scan.nextInt();**

**String choice1 = Integer.toString(choice);**

**oup.writeBytes(choice1);**

**oup.writeByte('\n');**

**int i;**

**String tokens[] = null;**

**String[] tokens1=null;**

**switch (choice)**

**{**

**case 1 :**

**System.out.println("Register:");**

**System.out.println("Enter The file name");**

**Scanner x=new Scanner(System.in);**

**String path=x.nextLine();**

**String files;**

**String fileinfo=words[0]+" "+port+" "+path;**

**oup.writeBytes(path);**

**oup.writeByte('\n');**

**oup.writeBytes(fileinfo);**

**oup.writeByte('\n');**

**/\* File folder = new File(path);**

**File[] listOfFiles = folder.listFiles();**

**int z=listOfFiles.length;**

**String str1 = Integer.toString(z);**

**oup.writeBytes(str1);**

**oup.writeByte('\n');**

**for (i = 0; i < z; i++)**

**{**

**if (listOfFiles[i].isFile())**

**{**

**files = listOfFiles[i].getName();**

**oup.writeBytes(files);**

**oup.writeByte('\n');**

**oup.writeBytes(fileinfo);**

**oup.writeByte('\n');**

**}**

**}\*/**

**String ip21 = inp.readLine();**

**System.out.println(ip21);**

**Socket r=new Socket(words[0],port+1111);**

**System.out.println("Peer1 sending file....");**

**DataInputStream inp2 = new DataInputStream(r.getInputStream());**

**DataOutputStream oup2 = new DataOutputStream(r.getOutputStream());**

**String save="upload";**

**oup2.writeBytes(save);**

**oup2.writeByte('\n');**

**File myFile = new File (path);**

**byte [] buffer1 = new byte [(int)myFile.length()];**

**FileInputStream fis = new FileInputStream(myFile);**

**BufferedInputStream bis = new BufferedInputStream(fis);**

**bis.read(buffer1,0,buffer1.length);**

**OutputStream os = r.getOutputStream();**

**System.out.println("Sending...");**

**os.write(buffer1,0,buffer1.length);**

**os.flush();**

**r.close();**

**break;**

**case 2 :**

**System.out.println("Enter filename to be search:");**

**String fname=scan.next();**

**oup.writeBytes(fname);**

**oup.writeByte('\n');**

**String ip6 = inp.readLine();**

**if(ip6.equals("FileNotFound"))**

**{**

**System.out.println(ip6);**

**}**

**else**

**{**

**System.out.println("Value = "+ip6 );**

**String phrase = ip6;**

**String delims = "[ ]+";**

**tokens1 = phrase.split(delims);**

**for ( i = 0; i < tokens1.length; i++)**

**{ }**

**int filesize=266392;**

**int bytesRead;**

**int current = 0;**

**System.out.println(tokens1[0]);**

**System.out.println(tokens1[1]);**

**System.out.println(tokens1[2]);**

**if(port== Integer.parseInt(tokens1[1]))**

**{**

**oup.writeBytes(tokens1[2]);**

**oup.writeByte('\n');**

**BufferedReader inFromUser = new BufferedReader(new InputStreamReader( System.in));**

**System.out.println("connecting");**

**byte [] buffer = new byte [filesize];**

**InputStream is = s.getInputStream();**

**FileOutputStream fos = new FileOutputStream("wassupp.class");**

**BufferedOutputStream bos = new BufferedOutputStream(fos);**

**// current = is.read(buffer,0,buffer.length);**

**//current = bytesRead;**

**// System.out.println(current);**

**do {**

**bytesRead =is.read(buffer);**

**// current, (buffer.length-current));**

**bos.write(buffer, 0 , bytesRead);**

**System.out.println("file downloaded");**

**bos.flush();**

**//if(bytesRead >= 0)**

**//current += bytesRead;**

**System.out.println("-------------------");**

**} while(bytesRead >=0);**

**//bos.write(buffer, 0 , current);**

**//System.out.println("file downloaded");**

**//bos.flush();**

**bos.close();**

**}**

**else**

**{**

**Socket clientSocket = new Socket(tokens1[0], Integer.parseInt(tokens1[1]));**

**System.out.println("connecting");**

**oup.writeBytes("download");**

**oup.writeByte('\n');**

**oup.writeBytes(tokens1[2]);**

**oup.writeByte('\n');**

**BufferedReader inFromUser = new BufferedReader(new InputStreamReader( System.in));**

**System.out.println("connecting");**

**byte [] buffer = new byte [filesize];**

**InputStream is = s.getInputStream();**

**FileOutputStream fos = new FileOutputStream("wassupp.class");**

**BufferedOutputStream bos = new BufferedOutputStream(fos);**

**// current = is.read(buffer,0,buffer.length);**

**//current = bytesRead;**

**// System.out.println(current);**

**do {**

**bytesRead =is.read(buffer);**

**// current, (buffer.length-current));**

**bos.write(buffer, 0 , bytesRead);**

**System.out.println("file downloaded");**

**bos.flush();**

**//if(bytesRead >= 0)**

**//current += bytesRead;**

**System.out.println("-------------------");**

**} while(bytesRead >=0);**

**//bos.write(buffer, 0 , current);**

**//System.out.println("file downloaded");**

**//bos.flush();**

**bos.close();**

**clientSocket.close();**

**}**

**}**

**break;**

**default :**

**System.out.println("Wrong Entry ");**

**break;**

**}**

**System.out.println("Do you want to continue (Type y or n) \n");**

**ch = scan.next().charAt(0);**

**String str = Character.toString(ch);**

**oup.writeBytes(str);**

**oup.writeByte('\n');**

**} while (ch == 'Y'|| ch == 'y');**

**}**

**k++;**

**}**

**br.close();**

**}**

**catch (Exception e)**

**{ System.err.println("Error: " + e.getMessage());**

**}**

**}**

**}**

# Evaluation

# EvaluationIndexingserver.java

**import java.io.\*;**

**import java.net.\*;**

**import java.io.BufferedReader;**

**import java.io.BufferedWriter;**

**import java.io.DataInputStream;**

**import java.io.InputStreamReader;**

**import java.util.Scanner;**

**import java.util.Hashtable;**

**import java.util.ArrayList;**

**import java.util.HashMap;**

**import java.util.Iterator;**

**import java.util.Map;**

**import java.util.Map.Entry;**

**import java.util.Set;**

**class Hash**

**{**

**public static int currentSize, maxSize;**

**public static String keys;**

**public static String vals;**

**public static Hashtable<String,String> data= new Hashtable<String,String>(1000001);**

**public Hash()**

**{ currentSize = 0;**

**keys = new String();**

**vals = new String();**

**}**

**void insert(String key, String val)**

**{**

**keys=key;**

**vals=val;**

**data.put(keys,vals);**

**return;**

**}**

**public String get(String Name){**

**keys=Name;**

**return data.get(keys);**

**}**

**void delete(String key)**

**{**

**keys=key;**

**data.remove(keys);**

**}**

**void printHashTable()**

**{ System.out.println("Hash Table " );**

**Iterator<Entry<String, String>> it = data.entrySet().iterator();**

**while (it.hasNext())**

**{**

**Entry<String, String> pair = it.next();**

**System.out.println(pair.getKey() + " " + pair.getValue());**

**}**

**}**

**}**

**class ThreadHandler extends Thread**

**{**

**Socket News;**

**int n;**

**ThreadHandler(Socket s,int v)**

**{**

**News=s;**

**n=v;**

**}**

**public void run()**

**{**

**try**

**{**

**System.out.println("Thread created for peer" );**

**Scanner scan = new Scanner(System.in);**

**DataInputStream inp = new DataInputStream(News.getInputStream());**

**DataOutputStream oup = new DataOutputStream(News.getOutputStream());**

**Hash h1 = new Hash();**

**char ch;**

**do**

**{**

**String ip = inp.readLine();**

**int choice2 = Integer.parseInt(ip);**

**switch (choice2)**

**{**

**case 1 :**

**// String ip31 = inp.readLine();**

**// int q = Integer.parseInt(ip31);**

**for(int k=10000;k<20000;k++)**

**{**

**h1.insert(inp.readLine(), inp.readLine() );**

**}**

**String ip15="Success";**

**oup.writeBytes(ip15);**

**oup.writeByte('\n');**

**System.out.println("Files are inserted");**

**break;**

**case 2 :**

**for(int k=10000;k<20000;k++)**

**{**

**String ip3 = inp.readLine();**

**String ip11=(String) h1.get(ip3);**

**if(ip11==null)**

**{ String i26="FileNotFound";**

**oup.writeBytes(i26);**

**oup.writeByte('\n');**

**}**

**else**

**{oup.writeBytes(ip11);**

**oup.writeByte('\n');**

**}**

**}**

**BufferedReader inFromClient = new BufferedReader(new InputStreamReader(News.getInputStream()));**

**DataOutputStream outToClient = new DataOutputStream( News.getOutputStream());**

**for(int k=10000;k<20000;k++)**

**{**

**String Dfile = inp.readLine();**

**File myFile = new File (Dfile);**

**byte [] buffer = new byte [(int)myFile.length()];**

**FileInputStream fis = new FileInputStream(myFile);**

**BufferedInputStream bis = new BufferedInputStream(fis);**

**bis.read(buffer,0,buffer.length);**

**OutputStream os = News.getOutputStream();**

**System.out.println("Sending...");**

**os.write(buffer,0,buffer.length);**

**os.flush();**

**}**

**break;**

**default :**

**System.out.println("Wrong Entry ");**

**break;**

**}**

**h1.printHashTable();**

**String ctr = inp.readLine();**

**ch = ctr.charAt(0);**

**} while (ch == 'Y'|| ch == 'y');**

**// News.close();**

**}**

**catch(Exception e)**

**{**

**System.out.println(e);**

**}**

**}**

**}**

**public class EvaluationIndexingServer**

**{**

**public static void main(String[] args)**

**{**

**int req=1001;**

**try**

**{**

**System.out.println("Enter The port of tthe server:");**

**Scanner x=new Scanner(System.in);**

**String port1=x.nextLine();**

**int port = Integer.parseInt(port1);**

**ServerSocket ss=new ServerSocket(port);**

**for(;;)**

**{**

**Socket s=ss.accept(); //establishes connection**

**System.out.println("Server started ");**

**Thread T =new ThreadHandler(s,req);**

**T.start();**

**req++;**

**}**

**}**

**catch(Exception e)**

**{System.out.println(e);}**

**}**

**}**

**B.EvaluationPeerClient.java**

**import java.io.\*;**

**import java.net.\*;**

**import java.io.BufferedReader;**

**import java.io.BufferedWriter;**

**import java.io.DataInputStream;**

**import java.io.File;**

**import java.io.FileInputStream;**

**import java.io.FileWriter;**

**import java.io.InputStreamReader;**

**import java.util.Scanner;**

**class Hashtableop**

**{**

**private int maxSize;**

**private String[] keys;**

**public Hashtableop(int capacity)**

**{**

**maxSize = capacity;**

**keys = new String[maxSize];**

**}**

**private int hash(String key)**

**{**

**return key.hashCode() % maxSize;**

**}**

**public int find(String key)**

**{**

**int tmp = hash(key);**

**return tmp;**

**}**

**}**

**class EvaluationPeerClient {**

**public static void main(String args[])**

**{**

**try**

**{**

**Scanner scan = new Scanner(System.in);**

**Hashtableop Serverslct = new Hashtableop(8);**

**System.out.println("Enter key to connect server:");**

**int n=Serverslct.find(scan.next());**

**if (n < 0)**

**n = -n;**

**System.out.println("connecting to the server "+n);**

**int k=0;**

**BufferedReader br = new BufferedReader(new InputStreamReader(new FileInputStream("Config.txt")));**

**String Peerdetls;**

**String line;**

**while ((line = br.readLine()) != null)**

**{**

**if(n==k)**

**{**

**Peerdetls=line;**

**String words[] = Peerdetls.split(" ");**

**// String firstTwo = words[0] + " " + words[1];**

**System.out.println(words[0]);**

**System.out.println(words[1]);**

**int port = Integer.parseInt(words[1]);**

**Socket s=new Socket(words[0],port);**

**System.out.println("Peer1 Intitialized");**

**DataInputStream inp = new DataInputStream(s.getInputStream());**

**DataOutputStream oup = new DataOutputStream(s.getOutputStream());**

**char ch;**

**do**

**{**

**System.out.println("\nHash Table Operations\n");**

**System.out.println("1. Register ");**

**System.out.println("2. Search & Download");**

**System.out.println("Enter the Choice:");**

**int choice = scan.nextInt();**

**String choice1 = Integer.toString(choice);**

**oup.writeBytes(choice1);**

**oup.writeByte('\n');**

**int i;**

**String tokens[] = null;**

**String[] tokens1=null;**

**switch (choice)**

**{**

**case 1 :**

**System.out.println("Register:");**

**System.out.println("Enter The file name");**

**Scanner x=new Scanner(System.in);**

**String path=x.nextLine();**

**String files;**

**String fileinfo=words[0]+" "+port+" "+path;**

**long millis = System.currentTimeMillis() % 1000;**

**for( i=10000;i<20000;i++)**

**{ String path1=i+path;**

**oup.writeBytes(path1);**

**oup.writeByte('\n');**

**oup.writeBytes(fileinfo);**

**oup.writeByte('\n');**

**}**

**long millis1 = System.currentTimeMillis() % 1000;**

**long Registertime=millis1-millis;**

**System.out.println("Time required to register 10k files"+Registertime);**

**String ip21 = inp.readLine();**

**System.out.println(ip21);**

**/\* File folder = new File(path);**

**File[] listOfFiles = folder.listFiles();**

**int z=listOfFiles.length;**

**String str1 = Integer.toString(z);**

**oup.writeBytes(str1);**

**oup.writeByte('\n');**

**for (i = 0; i < z; i++)**

**{**

**if (listOfFiles[i].isFile())**

**{**

**files = listOfFiles[i].getName();**

**oup.writeBytes(files);**

**oup.writeByte('\n');**

**oup.writeBytes(fileinfo);**

**oup.writeByte('\n');**

**}**

**}\*/**

**/\* Socket r=new Socket(words[0],port+1111);**

**System.out.println("Peer1 sending file....");**

**DataInputStream inp2 = new DataInputStream(r.getInputStream());**

**DataOutputStream oup2 = new DataOutputStream(r.getOutputStream());**

**String save="upload";**

**oup2.writeBytes(save);**

**oup2.writeByte('\n');**

**File myFile = new File (path);**

**byte [] buffer1 = new byte [(int)myFile.length()];**

**FileInputStream fis = new FileInputStream(myFile);**

**BufferedInputStream bis = new BufferedInputStream(fis);**

**bis.read(buffer1,0,buffer1.length);**

**OutputStream os = r.getOutputStream();**

**System.out.println("Sending...");**

**os.write(buffer1,0,buffer1.length);**

**os.flush();**

**r.close(); \*/**

**break;**

**case 2 :**

**System.out.println("Enter filename to be search:");**

**String fname=scan.next();**

**long millis2 = System.currentTimeMillis() % 1000;**

**for( i=10000;i<20000;i++)**

**{ String fname9=i+fname;**

**oup.writeBytes(fname9);**

**oup.writeByte('\n');**

**String ip6 = inp.readLine();**

**System.out.println("file details = "+ip6 );**

**}**

**long millis3 = System.currentTimeMillis() % 1000;**

**long searchtime=millis3-millis2;**

**System.out.println("Time required to search 10k files :"+searchtime);**

**String ip6 = inp.readLine();**

**if(ip6.equals("FileNotFound"))**

**{**

**System.out.println(ip6);**

**}**

**else**

**{**

**System.out.println("Value = "+ip6 );**

**String phrase = ip6;**

**String delims = "[ ]+";**

**tokens1 = phrase.split(delims);**

**for ( i = 0; i < tokens1.length; i++)**

**{ }**

**int filesize=266392;**

**int bytesRead;**

**int current = 0;**

**System.out.println(tokens1[0]);**

**System.out.println(tokens1[1]);**

**System.out.println(tokens1[2]);**

**long millis4 = System.currentTimeMillis() % 1000;**

**if(port== Integer.parseInt(tokens1[1]))**

**{**

**for( i=10000;i<20000;i++){**

**oup.writeBytes(tokens1[2]);**

**oup.writeByte('\n');**

**BufferedReader inFromUser = new BufferedReader(new InputStreamReader( System.in));**

**System.out.println("connecting");**

**byte [] buffer = new byte [filesize];**

**InputStream is = s.getInputStream();**

**FileOutputStream fos = new FileOutputStream("F2/");**

**BufferedOutputStream bos = new BufferedOutputStream(fos);**

**// current = is.read(buffer,0,buffer.length);**

**//current = bytesRead;**

**// System.out.println(current);**

**do {**

**bytesRead =is.read(buffer);**

**// current, (buffer.length-current));**

**bos.write(buffer, 0 , bytesRead);**

**System.out.println("file downloaded");**

**bos.flush();**

**//if(bytesRead >= 0)**

**//current += bytesRead;**

**System.out.println("-------------------");**

**} while(bytesRead >=0);**

**//bos.write(buffer, 0 , current);**

**//System.out.println("file downloaded");**

**//bos.flush();**

**bos.close();**

**}**

**long millis6 = System.currentTimeMillis() % 1000;**

**long obtaintime=millis6-millis4;**

**System.out.println("Time required to Obtain 10k files :"+obtaintime);**

**}**

**else**

**{**

**Socket clientSocket = new Socket(tokens1[0], Integer.parseInt(tokens1[1]));**

**System.out.println("connecting");**

**oup.writeBytes("download");**

**oup.writeByte('\n');**

**oup.writeBytes(tokens1[2]);**

**oup.writeByte('\n');**

**BufferedReader inFromUser = new BufferedReader(new InputStreamReader( System.in));**

**System.out.println("connecting");**

**byte [] buffer = new byte [filesize];**

**InputStream is = s.getInputStream();**

**FileOutputStream fos = new FileOutputStream("wassupp.class");**

**BufferedOutputStream bos = new BufferedOutputStream(fos);**

**// current = is.read(buffer,0,buffer.length);**

**//current = bytesRead;**

**// System.out.println(current);**

**do {**

**bytesRead =is.read(buffer);**

**// current, (buffer.length-current));**

**bos.write(buffer, 0 , bytesRead);**

**System.out.println("file downloaded");**

**bos.flush();**

**//if(bytesRead >= 0)**

**//current += bytesRead;**

**System.out.println("-------------------");**

**} while(bytesRead >=0);**

**//bos.write(buffer, 0 , current);**

**//System.out.println("file downloaded");**

**//bos.flush();**

**bos.close();**

**clientSocket.close();**

**}**

**}**

**break;**

**default :**

**System.out.println("Wrong Entry ");**

**break;**

**}**

**System.out.println("Do you want to continue (Type y or n) \n");**

**ch = scan.next().charAt(0);**

**String str = Character.toString(ch);**

**oup.writeBytes(str);**

**oup.writeByte('\n');**

**} while (ch == 'Y'|| ch == 'y');**

**}**

**k++;**

**}**

**br.close();**

**}**

**catch (Exception e)**

**{ System.err.println("Error: " + e.getMessage());**

**}**

**}**

**}**

**c.EvaluationServer.java**

import java.io.\*;

import java.net.\*;

import java.util.Scanner;

class ThreadHandler extends Thread

{

Socket News1;

int n;

ThreadHandler(Socket s,int v)

{

News1=s;

n=v;

}

public void run()

{

try

{

DataInputStream inp1 = new DataInputStream(News1.getInputStream());

DataOutputStream oup1 = new DataOutputStream(News1.getOutputStream());

String fp = inp1.readLine();

if(fp.equals("upload"))

{

int filesize=266392;

int bytesRead;

int current = 0;

BufferedReader inFromUser = new BufferedReader(new InputStreamReader( System.in));

System.out.println("connected");

byte [] buffer = new byte [filesize];

InputStream is = News1.getInputStream();

FileOutputStream fos = new FileOutputStream("jay123.txt");

BufferedOutputStream bos = new BufferedOutputStream(fos);

// current = is.read(buffer,0,buffer.length);

//current = bytesRead;

// System.out.println(current);

do

{ bytesRead =is.read(buffer);

// current, (buffer.length-current));

bos.write(buffer, 0 , bytesRead);

System.out.println("file downloaded");

bos.flush();

//if(bytesRead >= 0)

//current += bytesRead;

System.out.println("-------------------");

} while(bytesRead >=0);

//bos.write(buffer, 0 , current);

//System.out.println("file downloaded");

//bos.flush();

bos.close();

}

if(fp.equals("download"))

{

BufferedReader inFromClient = new BufferedReader(new InputStreamReader(News1.getInputStream()));

DataOutputStream outToClient = new DataOutputStream( News1.getOutputStream());

String Dfile = inp1.readLine();

File myFile = new File (Dfile);

byte [] buffer = new byte [(int)myFile.length()];

FileInputStream fis = new FileInputStream(myFile);

BufferedInputStream bis = new BufferedInputStream(fis);

bis.read(buffer,0,buffer.length);

OutputStream os = News1.getOutputStream();

System.out.println("Sending...");

os.write(buffer,0,buffer.length);

os.flush();

}

}

catch(Exception e)

{

System.out.println(e);

}

}

}

class EvaluationServer1 {

public static void main(String args[]) throws Exception {

int req=101;

try

{

System.out.println("Enter The port of tthe server:");

Scanner x=new Scanner(System.in);

String port2=x.nextLine();

int port5 = Integer.parseInt(port2);

for(;;)

{

ServerSocket welcomeSocket = new ServerSocket(port5);

Socket connectionSocket = welcomeSocket.accept();

System.out.println("I m the Client server:");

Thread T =new ThreadHandler(connectionSocket,req);

T.start();

req++;

}

}

catch(Exception e)

{System.out.println(e);}

}

}